SQUARE D COMPANY
Dedicated to Growth • Committed to Quality

GROUPE SCHNEIDER
SCHNEIDER NORTH AMERICA

REDACTED

DESIGN SPECIFICATION

Project Name: Circuit Monitor 3000/4000 Ethernet Communications Card (ECC)

Distribution:

Product Management	Design Engineering	Quality Assurance	Engineering Services
Rob Smits	Mike Pyle	Theresa Wall	Mark Hall
Imad Anayah	Jim Graff	Mike Divinnie	Bruce Ford
Don McComas	Jason Thurmond	Tim Slatton	
Jennifer Taylor	Jeff Farago	Victor Brown	
Daniel Langeac	Brooks Trace		
Bob Kennedy	Cedric Short		
	Brad Lazenby		
	Scott Northcutt		
	Jeff Fowler		

Approvals:

Marketing Engineer		Date:	
Development Engineer		Date:	
Product Manager		Date:	
Engineering Manager		Date:	

REDACTED

Rev	Revision Date	Revised by	Description
-	Thursday,	TGCurray	Initial Draft Release
1	Friday,	TGCurray	All references to CM3000 changed to CM3000/4000
2	Wednesday,	TGCurray	All references to EOM changed to ECC
3	Thursday,	TGCurray	Hardware sections updated to reflect the change to the non-5volt tolerant CPU
4	Tuesday,	TGCurray	Major overhaul – included items were related to: shielded twisted pair, cost updated, hardware real-estate/layout updated, power consumption updated, password lengths increased, minor HTML page changes, MMS functionality changed to potential, inter-processor communications reworked, manufacturing parameters reworked, environmental changes, register list updated.
5	Wednesday,	TGCurray	The key to components used in the Mechanical Feasibility figure was added.
6	Friday,	TGCurray	Another major overhaul – included items were related to: itemization of Phase I and Future release features updated, RS485 Interface distances tables updated, mechanical feasibility drawing figure updated, HTML page flow diagram figure updated, all HTML page figures updated to reflect Marketing feedback, more details about writing custom HTML pages added, inter-processor communications logistics updated, and register list updated to reflect the addition of manufacturing parameters.
7	Thursday,	TGCurray	More updates reflecting Product Marketing changes.
8	Monday,	TGCurray	Minor modifications to the inter-processor communications theory of operations.
9	Tuesday,	TGCurray	Modifications to the inter-processor communications boot-up sequence and the EEPROM byte offsets.
10	Thursday,	TGCurray	Modifications to the inter-processor communications theory of operations
11	Monday,	TGCurray	Modifications to the password HTML bit mapping table. Also the mechanical feasibility diagram was updated to reflect the RS485 wiring change.
12	Tuesday,	BALazenby	Modifications to the password HTML bit mapping table.
13	Tuesday,	TGCurray	Updated HTML pages to reflect Marketing changes. Also updated info in the RS485 section.
14	Thursday,	TGCurray	Updated the inter-processor communications section to reflect the use of the CM's RS485 port address for MBTCP communications.
15	Wednesday,	TGCurray JFowler	Updated the register list
16	Tuesday,	TGCurray	Updated the RS485 distances table to reflect

			testing results.

REDACTED

REDACTED

Table of Contents

REDACTED

REDACTED

List Of Tables

REDACTED

List Of Figures

REDACTED

1 Introduction

The Circuit Monitor 3000/4000 series Ethernet Communications Card (ECC) will be an installable option card, which can be utilized by the Square D CM3000/4000 series of meters (CM). This option card's primary function will be to allow the meters to be accessed over Ethernet media and provide a Gateway function by allowing Ethernet access to other Square D PowerLogic compatible Modbus, Jbus, and/or SyMax slave RS485 devices. This document will attempt to be a guide and provide the necessary information for the design of the hardware and firmware required for this device. The information in this document will be continually changing during the development phases and may be somewhat general in content pertaining to some areas of the design that will be finalized later during the development process. The sections labeled with (Future) are sections describing functionality that will be deferred until after the Phase I release.

1.1 General Functionality

The following is a list of most of the functionality that will be attempted for the ECC to support and if the functionality will be implemented in Phase I or later:

1	High-speed, direct Ethernet communications to the CM the ECC is inserted into	Phase I
2	RS485 support for up to 31 PowerLogic compatible, Modbus, Jbus, and/or SyMax slave devices	Phase I
3	RS485 2 and 4-wire communications with parity as even or none up to 38.4K baud	Phase I
4	10/100baseT Ethernet twisted pair (TP) communications support	Phase I
5	100baseFX Ethernet (Fiber) communications support	Phase I
6	Access to attached slave devices from client Modbus/TCP Ethernet communications	Phase I
7	ECC setup/configuration/diagnostics by password protected HTML Ethernet communications	Phase I
8	ECC firmware download by password protected FTP Ethernet communications	Phase I
9	HTML, custom device table download by password protected FTP Ethernet communications	Future
10	Multi-lingual HTML support	Future
11	ECC setup/configuration download by password protected FTP Ethernet communications	Future
12	Access to attached devices from client MMS/TCP Ethernet communications	Future
13	Time synchronization to attached slave devices by SNTP Ethernet communications	Future
14	Ability for CM/Sub-net master initiated communications	Future

Table 1 – ECC General Functionality

REDACTED

2 Hardware

The following shows the ECC electrical block diagram for the hardware layout:

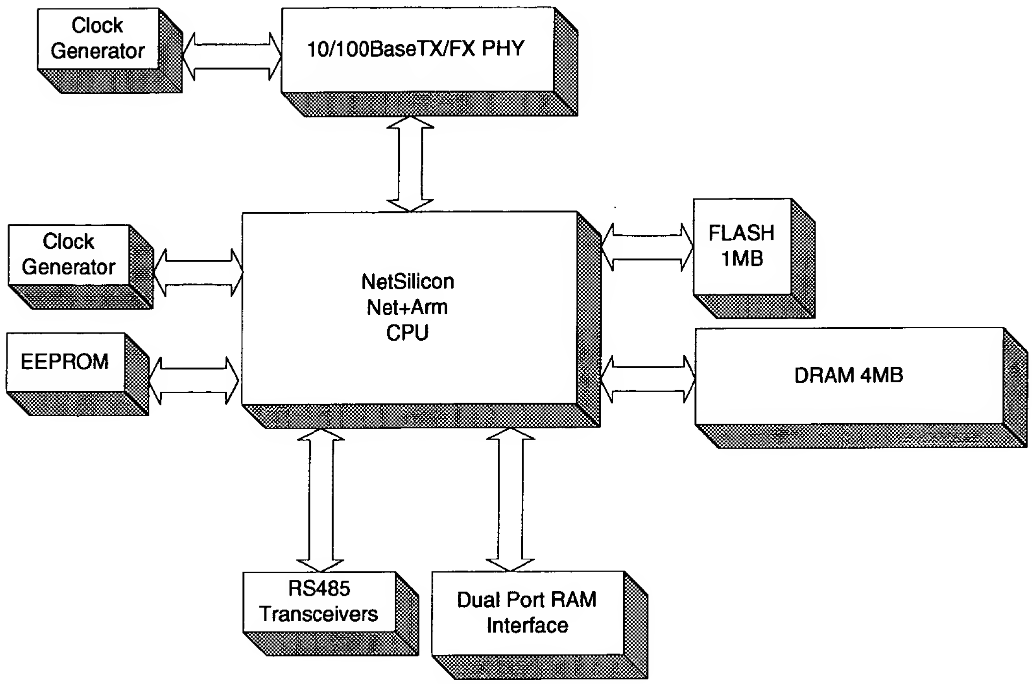


Figure 1 – ECC Electrical Block Diagram

2.1 Microprocessor

The microprocessor used for this design is the NETsilicon NET+ARM-40. This hardware architecture was developed by NETsilicon to optimize network performance. The microprocessor can run at 15MIPS (Million Instructions Per Second) but can actually reach 30MIPS peak performance with the correct peripherals and configuration of the onboard 4K cache. The internal frequency will be 33MHz driven by the external clock generator at 18.432MHz. The external bus speed will also be 33MHz. The following is a list of the main components contained in the NET+ARM chip that will be utilized in the design.

32bit 3.3V ARM7TDMI RISC Processor (not 5V tolerant)
10/100 Ethernet MAC
Glueless interface to 8, 16, and 32 bit peripherals
10 Channel DMA Controller
1 of 2 Asynchronous Serial Ports with FIFO
5 programmable chip selects
Programmable wait states
DRAM refresh controller
2 programmable timers
Programmable watchdog timer
Programmable Phase Lock Loop (PLL)
ARM embedded MULTI-ICE JTAG (not to be confused with 1149.1 JTAG Testing)

REDACTED

2.1.1 Chip Selects

The various peripheral devices are connected to one of the 5 programmable chip selects. These chip selects can be configured for 8, 16, or 32 bit peripherals. The following is a table of how the chip selects will be configured for use in this design.

Chip Select	Peripheral	Wait States	Start Address	Size	Width
CS0	Flash	1	0x0000000	1 Mbyte	16
CS1	DRAM RAS 1	0	0x1000000	4 Mbytes	32
CS2	-	-	-	-	-
CS3	EEPROM	4	0x2000000	8 Kbytes	8
CS4	Dual Port RAM	0	0x3000000	2 Kbyte	8

Table 2 – Chip Selects

2.1.2 Input/Output Pins

There are three ports of eight signals that can be configured to be used for either general purpose I/O bits, asynchronous serial channel signals, or interrupt inputs. Each port's signals can be configured independently. The following is a table of how the ports' signals will be used.

Signal	Type	Description
PortA0	-	Not used
PortA1	-	Not used
PortA2	-	Not used
PortA3	RXDA	Receive data serial channel A signal
PortA4	-	Not used
PortA5	RTSA	Request to send serial channel A signal
PortA6	-	Not used
PortA7	TXDA	Transmit data serial channel A signal
PortB0	-	Not used
PortB1	-	Not used
PortB2	-	Not used
PortB3	-	Not used
PortB4	-	Not used
PortB5	-	Not used
PortB6	-	Not used
PortB7	-	Not used
PortC0	Interrupt Input	External interrupt input signal for the Ethernet PHY
PortC1	Output	Output signal for serial channel A transmit LED
PortC2	Output	Output signal for serial channel A receive LED
PortC3	Interrupt Input	External interrupt input signal for the dual port RAM
PortC4	HDRS	Hardware reset output signal for soft resets
PortC5	Input	Optional Ethernet fiber transceiver detection input signal
PortC6	Input	Ethernet link input status signal
PortC7	-	Not used

Table 3 – Input/Output Pins

REDACTED

2.1.3 Fast Media Access Controller (MAC)

The NET+ARM has an integrated 10/100Mbit Media Access Controller (MAC) for Media Independent Interfaces (MII). This interface will be utilized in conjunction with an external Fast Ethernet PHY (Physical Interface) to allow for 10/100Mbit TP and 100Mbit Fiber communications.

2.2 Memory

Memory is commonly referred to as volatile and non-volatile types. The ECC will be designed using memory peripherals of both types. It will have 4Mbytes of DRAM and 2Kbytes of Dual Port RAM. Both of these peripherals are of the volatile memory type. It will also have 1Mbytes of Flash and 8Kbytes of EEPROM. Both of these peripherals are of the non-volatile memory type. The following table shows the memory ranges for the peripherals used in the design of the ECC.

Peripheral	Type	Size	Memory Range
Program Storage Flash	Non-Volatile	1Mbytes	0x0000000 – 0x00F9FFF
Program Execution/Stack DRAM	Volatile	4Mbytes	0x1000000 – 0x13E7FFF
Configuration storage EEPROM	Non-Volatile	8Kbytes	0x2000000 – 0x2001FFF
Inter-processor Communications DPRAM	Volatile	2Kbytes	0x3000000 – 0x30007FF

Table 4 – Memory Map

2.2.1 Flash

The Flash peripheral will be used for program (firmware) storage. At boot-up, the program code will be copied into DRAM for fast execution. The Flash will be an 8Mbit (512K x 16) part with an access time of 90ns.

2.2.2 DRAM

The DRAM peripherals will be used for program execution, stack space, and firmware downloads. There will be two 16Mbit (1M x 16) DRAM peripherals with access times of 50ns. These two peripherals will be used together to allow for 32bit wide data bus accesses by the processor. This combination of bus size and fast access speeds will allow for maximum execution of the processor with zero wait states.

2.2.3 EEPROM

The EEPROM peripheral will be used for limited configuration storage relevant to each particular ECC. Probably, the most important information will be the Ethernet Media Access Controller (MAC) address. The EEPROM will be a 64Kbit (8K x 8) part with an access time of 200ns.

2.2.4 Dual Port RAM

The dual port RAM peripheral will be used for the high-speed inter-processor communications between the ECC and the CM3000/4000 series devices. The dual port RAM will be a 16Kbit (2K x 8) part with an access time of 55ns. This speed will allow us to access this peripheral with no wait states at the full speed of the processor.

2.3 Other Peripherals and Components

There are a few other peripherals and components that are required to complete the ECC. These peripherals complement the processor and memory to allow for the fulfillment of all the functional requirements needed.

2.3.1 RS485 Interface

REDACTED

The RS485 communications interface will be a 4-wire plus shield interface (Tx+, Tx-, Rx+, Rx-, and Shld). This interface will have electrical isolation to internal circuitry up to 7500Vac(pk) for duration of one second or less by means of optical isolation and DC-to-DC converter. The transceiver used is protected against 15kV electrostatic discharge (ESD) shocks using the Human Body Model. Also, the transceiver features reduced slew-rate drivers that minimize Electromagnetic Interference (EMI) and reduce reflections caused by improperly terminated cables.

The connector used to wire into this interface will be a 5-point screw type commonly known as a "Phoenix" or "Terminal Block" connector (same as the one used on Power Meter devices).

Like all RS485 daisy chains, correct biasing is required to ensure reliable communications. Traditionally, a Multipoint Communications Adapter (MCA) is used at the beginning/master of the daisy chain. This adapter circuitry will be built into the ECC so no adapter will be needed externally. This internal biasing is calculated based on the daisy chain always being terminated and on using cabling with a 120 ohm impedance characteristic (Belden cable 9842). Also, RS485 daisy chain termination is required to ensure reliable communications. The last device on the daisy chain usually needs to have a Multipoint Communications Terminator (MCT). Ideally, the last device on the daisy chain should have a termination resistor of 120 ohms across its receive plus and minus pair only.

The RS485 interface will be designed to support up to 31 slave RS485 4-wire devices. The "guaranteed" maximum number of devices capable of being supported on a single daisy chain is determined based on the relation of baud rate, the length of the daisy chain, and the types of slave RS485 devices (2-wire/4-wire). The RS485 interface will support daisy chains that fall within the following specifications.

4-Wire*

Baud Rate	Max distance for 1-16 devices	Max distance for 17-32 devices
1200	10,000ft (3,048m)	10,000ft (3,048m)
2400	10,000ft (3,048m)	5,000ft (1,524m)
4800	10,000ft (3,048m)	5,000ft (1,524m)
9600	10,000ft (3,048m)	4,000ft (1,219m)
19200	5,000ft (1,524m)	2,500ft (762m)
38400	5,000ft (1,524m)	1,500ft (457m)

Table 5 – 4-wire RS485 Distances

2-Wire*

Baud Rate	Max distance for 1-8 devices	Max distance for 9-16 devices
1200	10,000ft (3,048m)	10,000ft (3,048m)
2400	10,000ft (3,048m)	5,000ft (1,524m)
4800	10,000ft (3,048m)	5,000ft (1,524m)
9600	10,000ft (3,048m)	4,000ft (1,219m)
19200	5,000ft (1,524m)	2,500ft (762m)
38400	2,500ft (1,524m)	1,500ft (457m)

Table 6 - 2-wire RS485 Distances

* Due to the volume of RS485 devices in the field, these tables are only to be used as a guide and were tabulated based on PowerLogic 4-wire devices and PowerLogic 4-wire devices which are capable of doing 2-wire.

2.3.2 Fast Ethernet Physical (PHY) Transceiver Interface

REDACTED

The Fast Ethernet Transceiver (PHY) provides a Media Independent Interface (MII) for easy attachment to the 10/100 Media Access Controller (MAC) which is integrated into the NET+ARM processor. The PHY is capable of directly driving an RJ45 interface through magnetics and termination resistors. The PHY also provides a pseudo-ECL interface for use with 100BaseFX fast fiber transceivers.

2.3.3 10/100BaseTX Interface (RJ45)

The RJ45 interface uses two signal pairs (one for transmit and one for receive) and a center tap for the transmit transformer. These same signal pairs, magnetics, and termination resistors are used for both 10Mbit and 100Mbit operation. This interface can drive a twisted pair cable up to 100m (328ft) in length when using data grade twisted-pair wire that has a characteristic impedance of 100 ohms and meets the EIA/TIA Category Five standard wire specifications.

The cable used can be either shielded twisted pair (STP) or unshielded twisted pair (UTP). Great care should be taken here to not use IBM type 1 cabling, which is STP at 150 ohm. In the past, STP meant IBM 150 ohm cabling. Today, there is Cat 5 shielded cabling which is 100 ohm. In the USA, the cable type used is usually unshielded, in Europe it is often shielded. Most shielded 4pair cables used today are 100 ohm, either with overall foil shield (FTP) or individually shielded pairs within a braided sheath (ScTP). Most of the industry appears to be going to ScTP.

The RJ45 interface is capable of auto negotiation for speed and duplex mode. If the link partner is also capable of auto negotiation, the two devices will exchange Fast Link Pulse (FLP) bursts to communicate their capabilities to each other. The highest common capabilities of the two will then be agreed upon. If the link partner is not capable of auto negotiation, the partner will be transmitting either 10Mbits Normal Link Pulses (NLP) or 100Mbit idle symbols. The RJ45 interface will detect either NLPs or Idle symbols and will automatically configure itself to match the speed but only in half-duplex mode.

The magnetics used here are determined by the requirements of the Fast Ethernet Physical (PHY) transceiver. The required magnetics will be a transformer module with a "transmit turns ratio" of 1:1.

2.3.4 Fast Fiber Transceiver

The Fast Fiber Transceiver is compliant with the optical performance requirements of the physical layer of the 100BaseFX version of the IEEE 802.3u specifications. This specification is defined as the FDDI PMD Standard ISO/IEC 9314-3:1990 and ANSI X3.166-1990.

The transceiver has a duplex LC connector receptacle and is compatible with 1300nm wavelength multimode fiber connections. It is optimized for 62.5 or 50/125 micron multimode graded index glass optical fiber per TIA-568A and ISO 11801. Also, the transceiver is capable of signal integrity up to 2000m in length of multimode full duplex fiber.

A few things to note, there is no industry standard for auto-negotiation on 100BaseFX. To use this interface, the user will have to force this mode by means of the setup interface. During development though, we may be able to possibly come up with a way to auto-negotiate to 100BaseFX with logic in the Firmware. Also, this interface does not support 10FL (10Mbits fiber) applications.

This component is the most expensive one in the design. The expense is so great that the board will be designed to have the ability to detect if this component is present or not. This will allow for the manufacture of two types of boards if needed. The primary board will have the fast fiber transceiver populated and will allow for the use of the twisted pair or the fiber interface. The secondary board will not have the fast fiber transceiver populated and will allow for the use of only the twisted pair interface.

REDACTED

2.3.5 Light Emitting Diodes (LEDs)

There will be two tri-level LED indicators. One will be of color green/green/yellow (Tri-level 1), and the other one will be of color yellow/green/yellow (Tri-level 2). The following table shows what each individual LED represents.

Unit and Color	Description
Tri-level 1 green	Power is being received from the source (CM)
Tri-level 1 green	RS485 Transmit in progress
Tri-level 1 yellow	RS485 Receive in progress
Tri-level 2 yellow	Ethernet link good
Tri-level 2 green	Ethernet Transmit in progress
Tri-level 2 yellow	Ethernet Receive in progress

Table 7 – LED Descriptions

2.3.6 Circuit/Processor Reset

The overall circuit board power is kept in check by a voltage monitor. This device monitors three vital system conditions. It monitors the 5-volt supply from the CM, the 3.3-volt supply from the CM, and the external override/reset controlled by the CM.

When an out-of-tolerance condition is detected on either of the voltages, this device will reset the board. This capability helps ensure against firmware corruption and other "flaky" operations that occur due to out-of-tolerance power events.

Also, this device will allow an easy way for the CM to be able to control the board in the need of a reset.

REDACTED

2.4 Power Supply

The ECC will not have its own power supply. The ECC will receive its power requirements from the CM it is inserted into. The CM will need to supply +5Vdc and +3.3Vdc to the ECC through the Option slot connectors. The amount of the supply should be based on the following feasibility study.

Quantity	Description	Typical Power (mW)	Maximum Power (mW)
1	Fast Fiber LC Connector	165	165
1	IC, NETA 40-3	330	495
1	IC, Altima PHY	280.5	330
1	IC, 8 Mbit Flash Memory	49.5	99
2	IC, 1M x 16 (16-MBIT) DRAM	957	1056
1	IC, 8K x 8 EEPROM	0.165	26.4
1	IC, 1K x 8 Dual-Port RAM	247.5	297
1	Crystal, 18.432MHZ SMT		
1	Crystal, 25MHZ SMT		
1	Microprocessor Reset	0.25	0.25
3	Opto Isolator	375	600
1	DC/DC Converter	400	625
1	RS485 Dual Driver/Receiver	0.6	5
1	Single Port Xformer		
1	PCB RJ-45 Shielded Jack		
1	Tri-Level LED		
1	Tri-Level LED		
1	5 Position Terminal Plug		
1	5 Position Terminal Plug		
1	Connector 48Pin Eurocard		
1	Misc Passive/PCB	49.5	108.9
	Twisted Pair Only Total >	*2690.015*	*3642.55*
	Total >	*2855.015*	*3807.55*
	3.3V Burden Total >	*2079.165*	*2577.3*
	5V Burden Total >	*775.85*	*1230.25*

Table 8 – ECC Estimated Power Consumption

REDACTED

2.5 Mechanical

The size and dimensions of the ECC has already been pre-determined by the specifications of the option slots in the CM. The following is a feasibility study of what the final board will potentially look like.

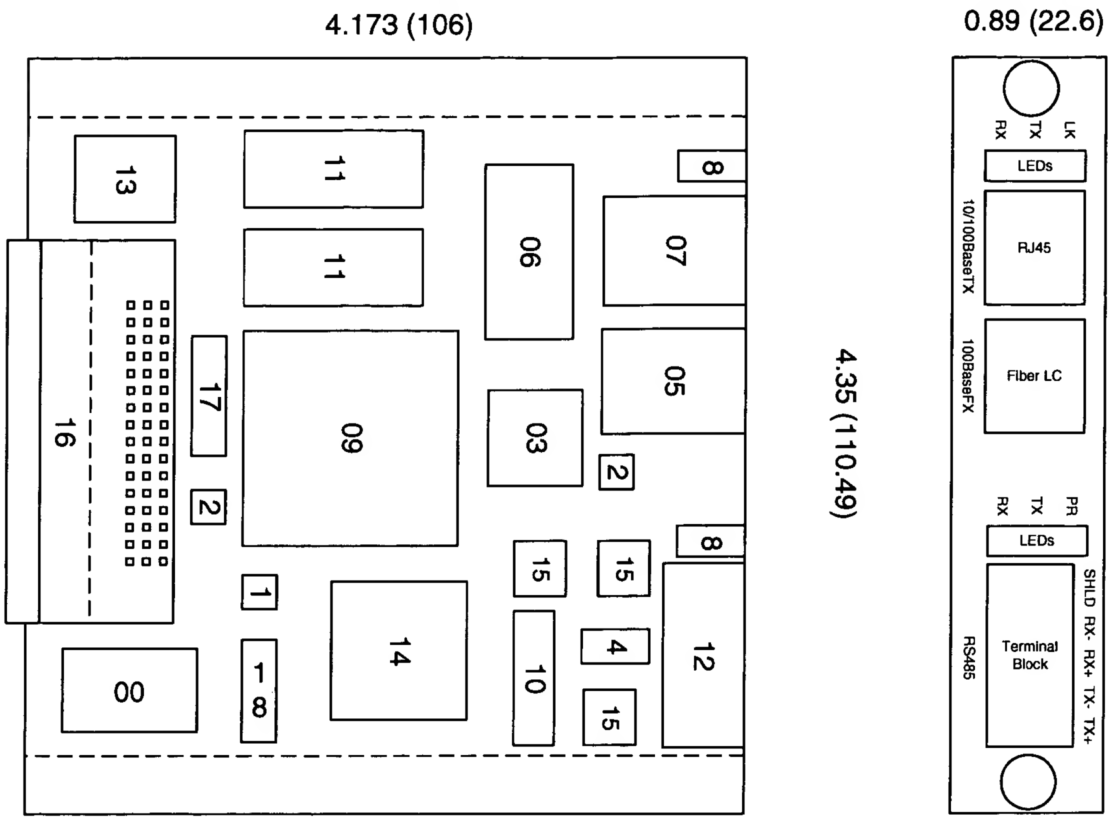


Figure 2 – Mechanical Feasibility Drawing

Key #	Quantity	Component	Key #	Quantity	Component
00	1	Flash	10	1	DC to DC
01	1	Reset	11	2	Dram
02	2	Crystal	12	1	Terminal Block
03	1	PHY	13	1	EEProm
04	1	RS485	14	1	Dual Port Ram
05	1	Fiber	15	3	Optocouplers
06	1	Transformer	16	1	Eurocard Connector
07	1	RJ45	17	1	Jtag Connector
08	2	LEDs	18	1	Bus Switch
09	1	CPU			

Table 9 – Component Key for Mechanical Feasibility Drawing

REDACTED

2.6 ECC Estimated RAW Component Costs

The following is a breakdown listing based on raw quoted component costs.

Quantity	Description	Price Each	Total
1	Fast Fiber LC Transceiver		
1	IC, Net+Arm CPU		
1	IC, Altima PHY		
1	IC, 8 Mbit Flash Memory		
2	IC, 1M x 16 (16-MBIT) DRAM		
1	IC, 8K x 8 EEPROM		
1	IC, 2K x 8 Dual-Port RAM		
1	Crystal, 18.432MHZ SMT		
1	Crystal, 25MHZ SMT		
1	Microprocessor Reset		
3	Opto Isolator		
1	DC/DC Converter		
1	5V RS485 Dual Driver/Receiver		
1	Single Port Xformer		
1	PCB RJ-45 Shielded Jack		
1	Tri-level LED		
1	Tri-level LED		
1	5 Position Terminal Plug		
1	5 Position Terminal Plug		
1	Connector 48Pin Eurocard		
1	Misc Passives/PCB		
		Twisted Only >	
		Fiber Only >	
		Both Total >	

Table 10 – ECC Estimated RAW Component Costs

3 Ethernet Communications

Ethernet communications will be utilized, wherever possible and feasible to do so, to allow for the "remote" control/maintenance of the ECC. The goal of this design is to make the ECC a passive/reactive device requiring minimal setup. The ECC will basically react to the "outside" world and do what it "thinks" is the best thing to do unless configured to do something different via setup. The Ethernet capabilities will be primarily used for ECC setup, diagnostics, firmware update, and access to attached slave devices by means of Hypertext Markup Language (HTML) and Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), Modbus over TCP/IP (ModbusTCP), and future plans for Manufacturing Message Specification over TCP/IP (MMSTCP).

REDACTED

3.1 HTTP Server and HTML Pages

A small subset of HTML v1.0 will be used with the HTTP server in the ECC primarily for ECC setup and diagnostics. With the onset and popularity of the World Wide Web and the Internet, the use of a browser is practically commonplace. This capability of the ECC allows for almost any user of an Internet browser to easily access and configure the ECC device. The recommended browser of choice should be Internet Explorer v5.0 or greater.

3.1.1 HTML Page Flow

The following figure is what the ECC's HTML page flow will potentially be like.

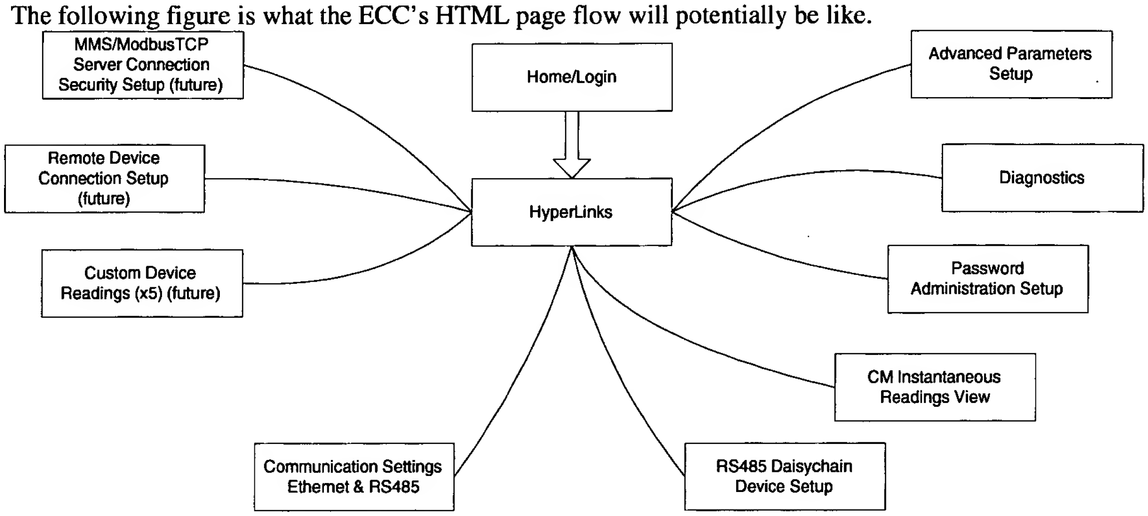


Figure 3 – HTML Page Flow Diagram

3.1.2 HTML Security

The ECC's HTML security will be designed in such a way to allow for four configurable account password access levels. These password access levels will be divided into three Password Accounts and one Administrator Account.

3.1.2.1 HTML Security Theory of Operation

When a user attempts to access the ECC for the first time during a web browser session, the ECC will force that user into the password logon page. Once there, the users will login to the ECC using one of the four defined passwords. During this login, the ECC will generate a random number and associate that password access level with it. The random number will then be the users "access token" and will automatically be appended to all the HTML page transactions during that session. This token will stay "alive" as long as the user keeps the session active by making requests to the ECC. Once this token is inactive for more than the default of ten minutes, the ECC will "expire" the token. The ECC will allow any combination, including redundant use, of all the passwords to allow up to ten access tokens to be active at any given time. This limits the number of active users viewing HTML pages to ten. The amount of time the ECC will wait during the inactivity period before "expiring" an access token will be configurable. For normal operations, it is recommended that the user return to the main links page and select to "log out" when fished interfacing with the ECC to immediately release that token slot for another user.

REDACTED

3.1.2.2 HTML Administrator Account

One of the four password accounts in the ECC will always be the Administrator password. This password account will always be granted full access to every HTML page available in the ECC and will also overlap to be the only password used for FTP operations. The only part of this account that is configurable is the password itself. The Administrator password will default to "admin". The Administrator password will be from one to eight characters and stored in four registers like the following table.

Register*	HiByte	LoByte
514	Admin password ASCII character 1	Admin password ASCII character 2
515	Admin password ASCII character 3	Admin password ASCII character 4
516	Admin password ASCII character 5	Admin password ASCII character 6
517	Admin password ASCII character 7	Admin password ASCII character 8

Table 11 – Administrator Password Setup Parameters

* Register numbers used in the tables throughout this document are CM registers.

3.1.2.3 HTML User Password Accounts

Three of the four password accounts in the ECC will be left to be used only for HTML access and will be called the User Password Accounts. Only the Administrator Account sets up these passwords. The parts of these accounts that are configurable are the passwords themselves and the level of access each password will allow to each HTML page in the ECC. These passwords will be configurable to allow no access, view-only access, or full access to each individual HTML page in the ECC. These four passwords can then be selectively given to and used by multiple users. The user passwords will default to "master", "engineer", and "operator". The passwords will be from one to eight characters and stored in four registers each like the following tables.

Register	HiByte	LoByte
518	Pass1 password ASCII character 1	Pass1 password ASCII character 2
519	Pass1 password ASCII character 3	Pass1 password ASCII character 4
520	Pass1 password ASCII character 5	Pass1 password ASCII character 6
521	Pass1 password ASCII character 7	Pass1 password ASCII character 8

Table 12 – Pass1 Password Setup Parameters

Register	HiByte	LoByte
526	Pass2 password ASCII character 1	Pass2 password ASCII character 2
527	Pass2 password ASCII character 3	Pass2 password ASCII character 4
528	Pass2 password ASCII character 5	Pass2 password ASCII character 6
529	Pass2 password ASCII character 7	Pass2 password ASCII character 8

Table 13 – Pass2 Password Setup Parameters

Register	HiByte	LoByte
534	Pass3 password ASCII character 1	Pass3 password ASCII character 2
535	Pass3 password ASCII character 3	Pass3 password ASCII character 4
536	Pass3 password ASCII character 5	Pass3 password ASCII character 6
537	Pass3 password ASCII character 7	Pass3 password ASCII character 8

Table 14 – Pass3 Password Setup Parameters

REDACTED

The access level each password will have to each HTML page will be held in four registers, totaling twelve. This means that each password will have four registers associated with it according to the following table.

Register	HiByte	LoByte
522	Pass1 Password HTML Page Access Bitmap (Most Significant Word)	
523	Pass1 Password HTML Page Access Bitmap (2nd Most Significant Word)	
524	Pass1 Password HTML Page Access Bitmap (2nd Least Significant Word)	
525	Pass1 Password HTML Page Access Bitmap (Least Significant Word)	
530	Pass2 Password HTML Page Access Bitmap (Most Significant Word)	
531	Pass2 Password HTML Page Access Bitmap (2nd Most Significant Word)	
532	Pass2 Password HTML Page Access Bitmap (2nd Least Significant Word)	
533	Pass2 Password HTML Page Access Bitmap (Least Significant Word)	
538	Pass3 Password HTML Page Access Bitmap (Most Significant Word)	
539	Pass3 Password HTML Page Access Bitmap (2nd Most Significant Word)	
540	Pass3 Password HTML Page Access Bitmap (2nd Least Significant Word)	
541	Pass3 Password HTML Page Access Bitmap (Least Significant Word)	

Table 15 – Passwords Access Setup Parameters

Within these registers, the bits are utilized to represent the access ability of each password to each HTML page. Each HTML page will be represented by two bits. This means that there can be up to thirty-two pages secured by one of four access levels. The following table shows the value representation of the access levels.

Bit Values	Access Level
0x00	No access
0x01	Reserved
0x02	View only access
0x03	Full access

Table 16 – Password Access Values

REDACTED

The following table shows the representation of the access levels that the passwords have to each HTML page.

Register	Bits	HTML page
522	15 and 14	Pass1 access to Home Page
522	13 and 12	Pass1 access to CM Instantaneous Readings View
522	11 and 10	Pass1 access to Custom Device Page 1 View (Future)
522	9 and 8	Pass1 access to Custom Device Page 2 View (Future)
522	7 and 6	Pass1 access to Custom Device Page 3 View (Future)
522	5 and 4	Pass1 access to Custom Device Page 4 View (Future)
522	3 and 2	Pass1 access to Custom Device Page 5 View (Future)
522	1 and 0	Pass1 access to Communication Settings Setup
523	15 and 14	Pass1 access to RS485 Daisy chain Device Setup
523	13 and 12	Pass1 access to Diagnostics
523	11 and 10	Pass1 access to MMS/ModbusTCP Server Security Setup (Future)
523	9 and 8	Pass1 access to Remote Device Connection Setup (Future)
523	7 and 6	Reserved
523	5 and 4	Reserved
523	3 and 2	Reserved
523	1 and 0	Reserved
524	15 and 14	Reserved
524	13 and 12	Reserved
524	11 and 10	Reserved
524	9 and 8	Reserved
524	7 and 6	Reserved
524	5 and 4	Reserved
524	3 and 2	Reserved
524	1 and 0	Reserved
525	15 and 14	Reserved
525	13 and 12	Reserved
525	11 and 10	Reserved
525	9 and 8	Reserved
525	7 and 6	Reserved
525	5 and 4	Reserved
525	3 and 2	Reserved
525	1 and 0	Reserved
530, 531, 532, 533, 538, 539, 540, 541		This format is duplicated for registers 530, 531, 532, 533 (Pass2 access) and 538, 539, 540, 541 (Pass3 access)

Table 17 – Password HTML Access Bitmaps

REDACTED

The following figure is what the Password Administration Setup HTML page will potentially look like. This page will be accessible by the administrator password only.

Password Administration

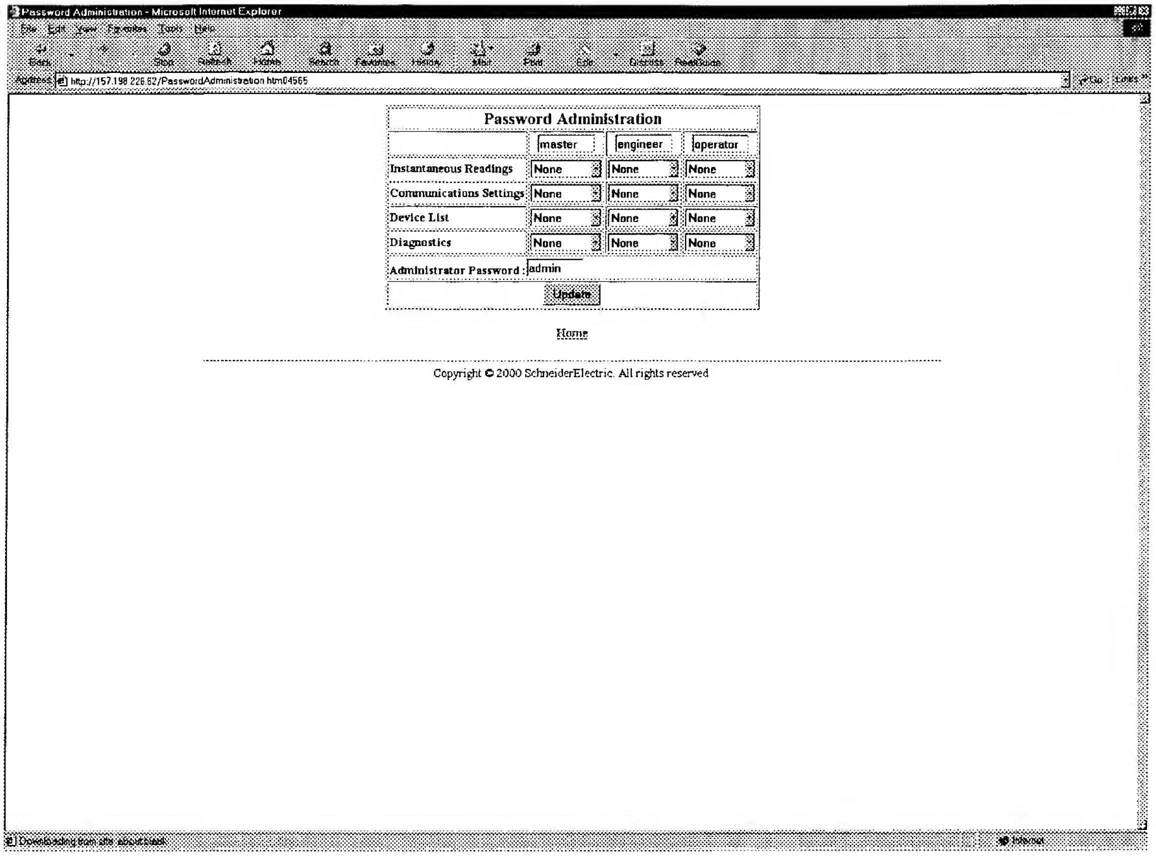


Figure 4 – Password Administration View

3.1.3 ECC Setup via HTML

All the setup information will be stored in the CM that the ECC is inserted into. This will allow for the ECCs to be "swappable". The ECC setup will primarily be done with HTML pages, but the CM display will have to be used during the initial steps.

3.1.3.1 Initial Setup

After the physical installation, the first step to completing the ECC setup will be via the CM display. The CM display will be used to setup the initial TCP/IP address, TCP/IP subnet mask, the TCP/IP router, and the Ethernet physical connection to use (fiber or twisted pair). With these parameters in place, the ECC will be accessible via the Ethernet, and the rest of the ECC setup can be done via HTML and a standard web browser or by FTP (basic setup only - Future).

REDACTED

.1.3.2 HTML Ethernet – TCP/IP Setup

After the one-time initial step of getting the TCP/IP address assigned to the ECC through the CM display, the ECC can, from that point on, have its TCP/IP setup changed with HTML pages and a standard web browser. The following is the information used for the TCP/IP setup.

Register	HiByte	LoByte
500	IP Address 1st Octet (0 – 255)	IP Address 2nd Octet (0 – 255)
501	IP Address 3rd Octet (0 – 255)	IP Address 4th Octet (0 – 255)
502	IP Sub-net Mask 1st Octet (0 – 255)	IP Sub-net Mask 2nd Octet (0 – 255)
503	IP Sub-net Mask 3rd Octet (0 – 255)	IP Sub-net Mask 4th Octet (0 – 255)
504	IP Router Address 1st Octet (0 – 255)	IP Router Address 2nd Octet (0 – 255)
505	IP Router Address 3rd Octet (0 – 255)	IP Router Address 4th Octet (0 – 255)
506	Ethernet physical connection (0 = UTP, 1 = Fiber)	

Table 18 – TCP/IP Setup Parameters

REDACTED

The following figure is what the Ethernet TCP/IP Setup HTML page will potentially look like. The "Update" button will not be seen if the user has "view only" access.

Communications Settings

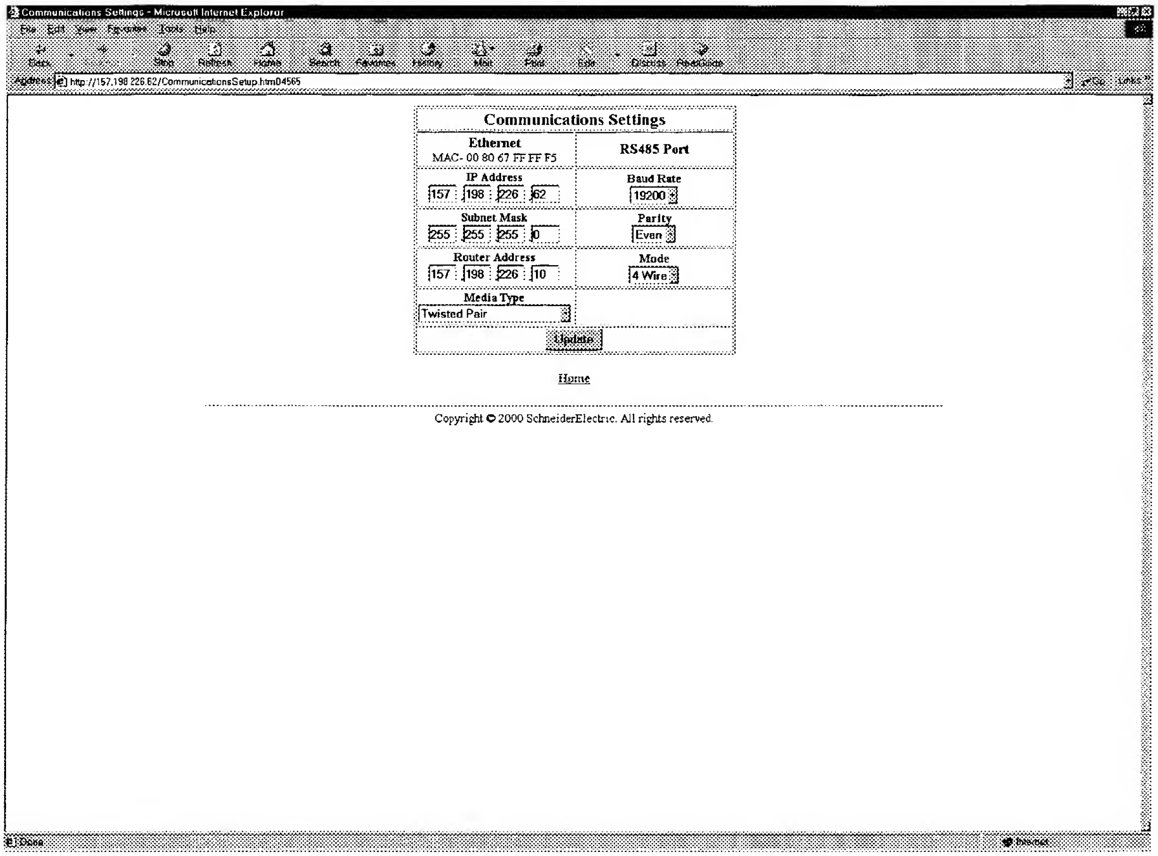


Figure 5 – Communications Settings View

3.1.3.3 HTML RS485 Device Definition Setup

REDACTED

Keeping in mind that we want the user to have to do as little setup as possible, all the devices on the daisy chain normally do not have to be defined. The ECC will be designed to function primarily as a passive Gateway. This means that the ECC will be able to pass through all Ethernet messages to the RS485 daisy chain based on the Ethernet protocol/format that the message was made. There are, however, scenarios in which this technique will not work, and the RS485 device must be defined so the ECC will be able to translate from one protocol to another. The devices that have to be selectively defined are PowerLogic protocol devices. PowerLogic protocol devices only have to be defined when a user wishes to communicate to them with the Ethernet protocol ModbusTCP. There will be a default list of 16 available device identification slots in the setup page interface for device definitions that will consist of an RS485 address and a protocol associated with it. The first entry will always show the slave address of the CM the ECC is inserted into. The following table shows the information used for defining a device.

Register	HiByte	LoByte
542	RS485 Device Definitions - Protocol (3 = PowerLogic, 8 = Modbus)	RS485 Device Definitions - Address (0 – 254)
543 – 604	62 more registers for up to 62 more Device definitions	
605	Number of viewable defined devices, includes CM that the ECC is attached to (2 – 62)	

Table 19 – RS485 Device Setup Parameters

One thing to note, the ECC will use the LoByte of these registers to build the route of 30,130,x to communicate to daisy-chained PowerLogic slave devices. Otherwise it will be used as the daisy-chained Modbus/Jbus slave ID.

REDACTED

The following figure is what the RS485 Device Setup HTML page will potentially look like. The "Update" button will not be seen if the user has "view only" access.

RS485 Device List Setup

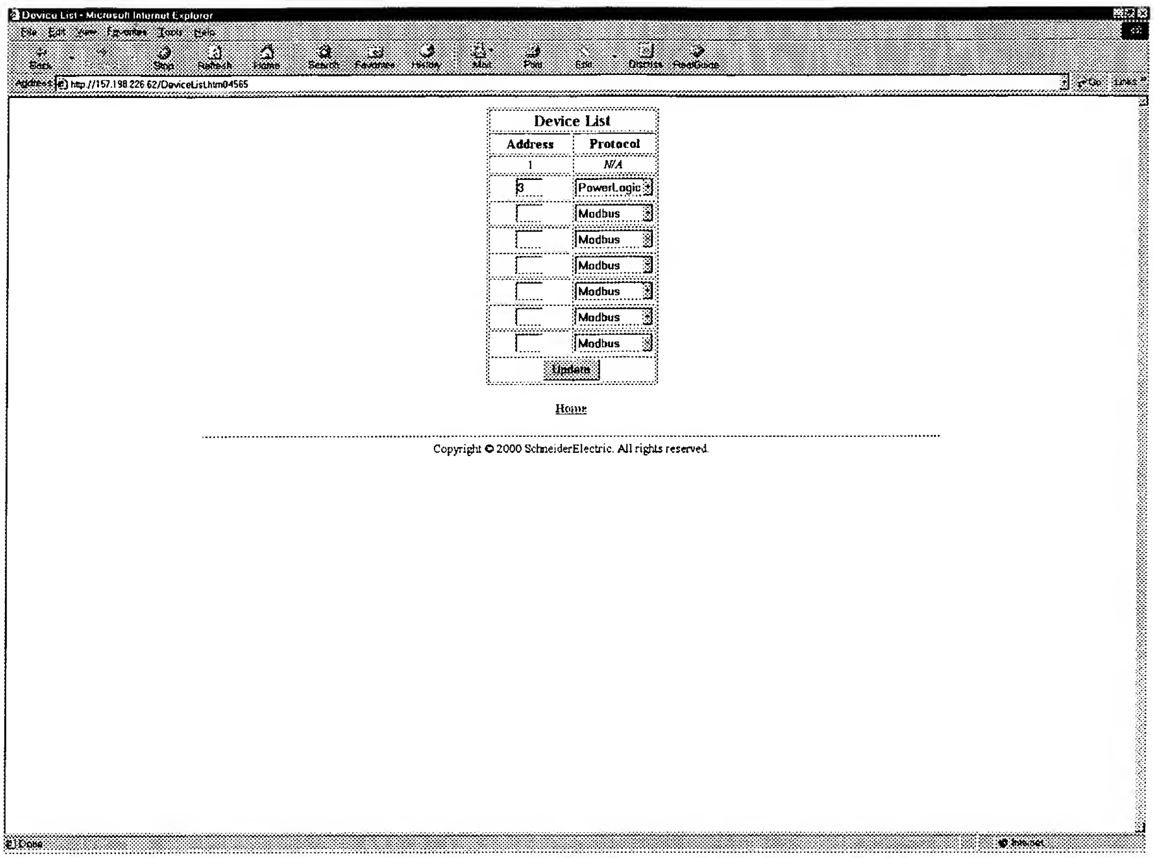


Figure 6 – RS485 Device List Setup View

3.1.3.4 HTML RS485 Setup

The RS485 Setup information will consist of baud rate, parity, and port mode. The following table shows the storage format of this setup information.

Register	HiByte	LoByte
512	RS485 Baud Rate (1200, 2400, 4800, 9600, 19200, 38400)	
513	RS485 Parity (0 = none, 2 = even)	RS485 Mode (0 = 4-wire Smart, 1 = 2-wire Smart)

Table 20 – RS485 Setup Parameters

REDACTED

The following figure is what the RS485 Serial Port Setup HTML page will potentially look like. The "Update" button will not be seen if the user has "view only" access.

Communications Settings

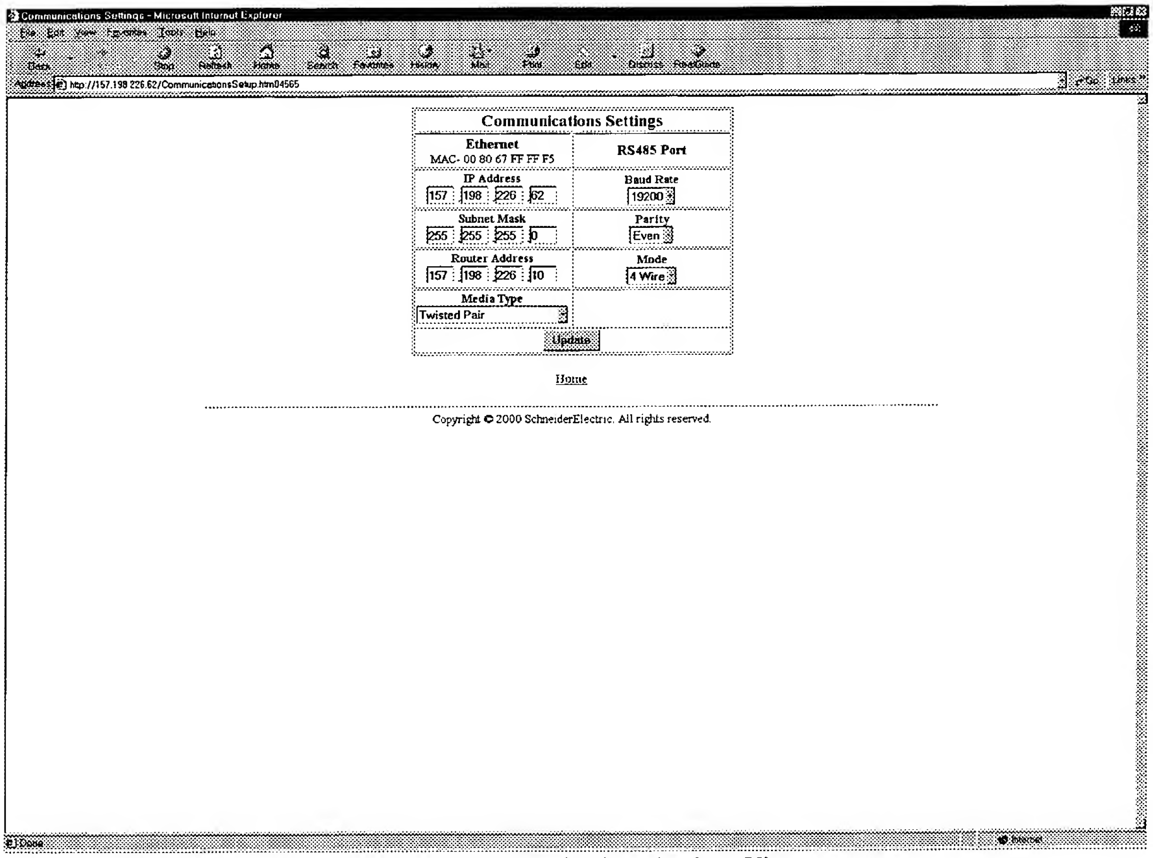


Figure 7 – Communications Settings View

REDACTED

3.1.3.5 HTML Remote Device Connection Setup (Future)

For sub-net initiated communications (the scenario where a serial master utilizes the ECC's RS485 port to gain access to remote locations/devices on the Ethernet, or the CM utilizes the ECC to gain access to remote locations/devices on the Ethernet), there has to be a way to define the remote nodes' locations. The ECC will have the ability to define up to ten remote Ethernet node locations. Also, the ECC will have the ability to define and associate up to eighty devices with these remote nodes. The following table is the format for the storage of the remote node/device information.

Registers	HiByte		LoByte
606	Associated Remote Node ID (0 – 9)	Remote Node Connection Type (1 = MMS SyMax format Port 1 2 = MMS SyMax format Port 2 3 = MMS Modbus format Port 1 4 = MMS Modbus format Port 2 5 = ModbusTCP)	Device Address 100's ID to be used in the Ethernet transaction (0 –247)
607 – 685	79 more registers for remote devices address 101 through 179		
686	Remote Node ID 0 IP Address 1st Octet (0 – 255)		Remote Node ID 0 IP Address 2nd Octet (0 – 255)
687	Remote Node ID 0 IP Address 3rd Octet (0 – 255)		Remote Node ID 0 IP Address 4th Octet (0 – 255)
688 – 705	18 more registers worth for remote node IDs 1 through 9		

Table 21 – Remote Device Connection Setup Parameters

The device initiating the communications will have to address to the ECC using addresses 100 through 179. These addresses will then serve as an identifier to "lookup" the information in that identifier's location. For Modbus initiators, the ECC will use the slave ID. For SyMax initiators, the ECC will use the last drop in the route. The information found in the location will then be used to make the Ethernet transaction. The following table shows the values for the connection types.

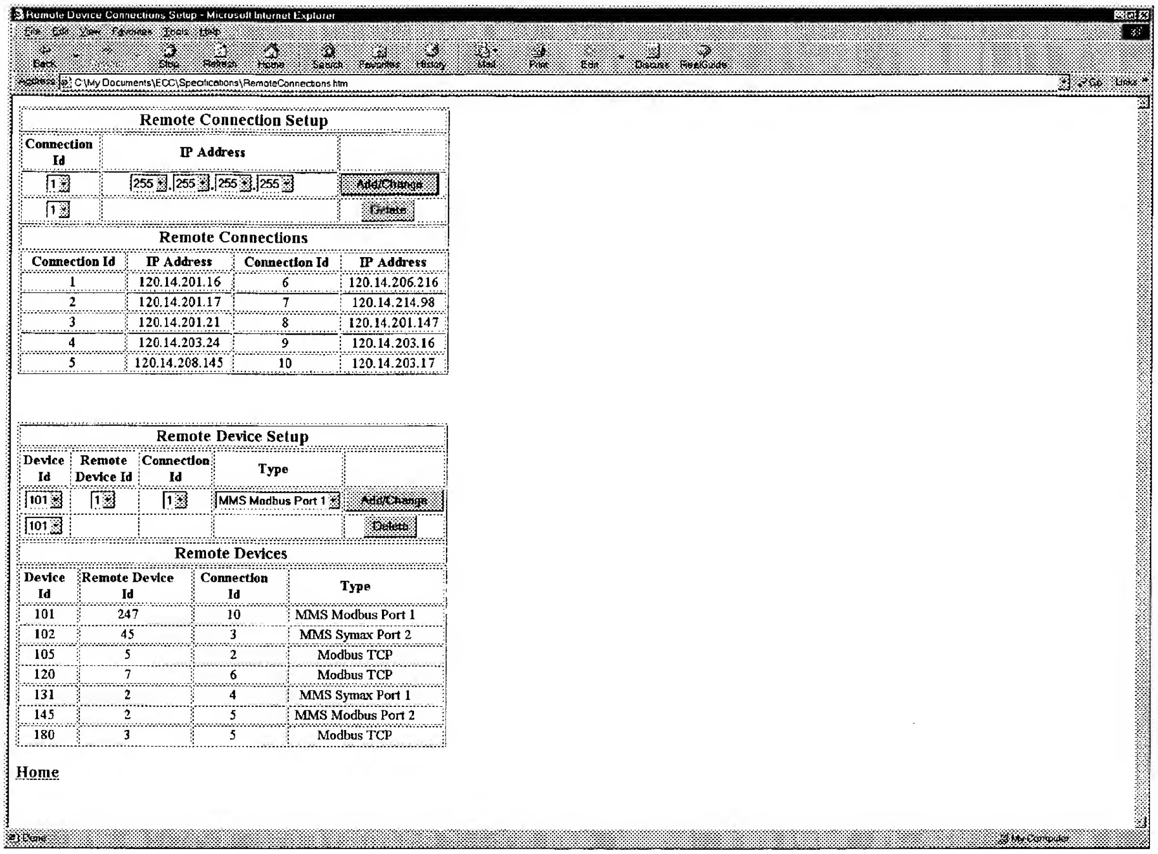
Remote Node Connection Type	Description
1	MMS SyMax format Port 1
2	MMS SyMax format Port 2
3	MMS Modbus format Port 1
4	MMS Modbus format Port 2
5	ModbusTCP

Table 22 – Remote Node Connection Types

The following figure is what the Remote Device Connections Setup HTML page will potentially look like. The Add/Change and Delete sections of the page will not be seen if the user has "view only" access.

REDACTED

Remote Device Connections Setup



Remote Connection Setup		
Connection Id	**IP Address**	
1	255 . 255 . 255 . 255	Add/Change
1		Delete

Remote Connections			
Connection Id	**IP Address**	**Connection Id**	**IP Address**
1	120.14.201.16	6	120.14.206.216
2	120.14.201.17	7	120.14.214.98
3	120.14.201.21	8	120.14.201.147
4	120.14.203.24	9	120.14.203.16
5	120.14.208.145	10	120.14.203.17

Remote Device Setup				
Device Id	**Remote Device Id**	**Connection Id**	**Type**	
101	1	1	MMS Modbus Port 1	Add/Change
101				Delete

Remote Devices			
Device Id	**Remote Device Id**	**Connection Id**	**Type**
101	247	10	MMS Modbus Port 1
102	45	3	MMS Symax Port 2
105	5	2	Modbus TCP
120	7	6	Modbus TCP
131	2	4	MMS Symax Port 1
145	2	5	MMS Modbus Port 2
180	3	5	Modbus TCP

Figure 8 – Remote Device Connections Setup View

REDACTED

3.1.3.6 HTML MMS/ModbusTCP Server Security Setup (Future)

The ECC can allow for the MMS/ModbusTCP server connections to have definable client access. This means that the IP addresses of the MMS/ModbusTCP clients will have to be defined in the ECC before the ECC will allow access for communications. This feature can be "turned on" in one of two ways. The first way is to turn it on so that all clients attempting to connect have read only access except for the defined clients that will have full access. The other way is to turn it on so all clients attempting to connect have no access except for the defined clients that will have a definable read only or full access. The ECC will have the ability to define ten clients.

Note: this mechanism works well until multiple clients access the ECC by way of a TCP/IP proxy server. In this scenario, all the clients would be connecting to the ECC with the same IP address (the one of the proxy server). Thus, the ECC would be checking access levels for multiple clients based on a single TCP/IP address.

The following table shows the format for the information for the client security access.

Register	HiByte	LoByte
706	Client IP Address 0 – 1st Octet (0 – 255)	Client IP Address 0 – 2nd Octet (0 – 255)
707	Client IP Address 0 – 3rd Octet (0 – 255)	Client IP Address 0 – 4th Octet (0 – 255)
708 - 725	18 more registers for client IP addresses 1 through 9	
726	Bits 15 and 14 signify how the security is turned on or if it is turned off. (00 = Security off 01 = Security on, all undefined clients have read only access and defined clients have automatic full access 10 = Security on, all undefined clients have no access and defined clients must have access level defined)	Bit 0 represents client 0, bit 1 represents client 1, through bit 9. (0 = Full Access, 1 = Read Only Access, these bits are ignored if bits 15 and 14 = 10)

Table 23 – MMS/Modbus TCP Security Setup Parameters

REDACTED

The following figure is what the MMS/ModbusTCP Security Setup HTML page will potentially look like. The "Update Settings" button will not be seen if the user has "view only" access.

MMS/MBTCP Security Setup

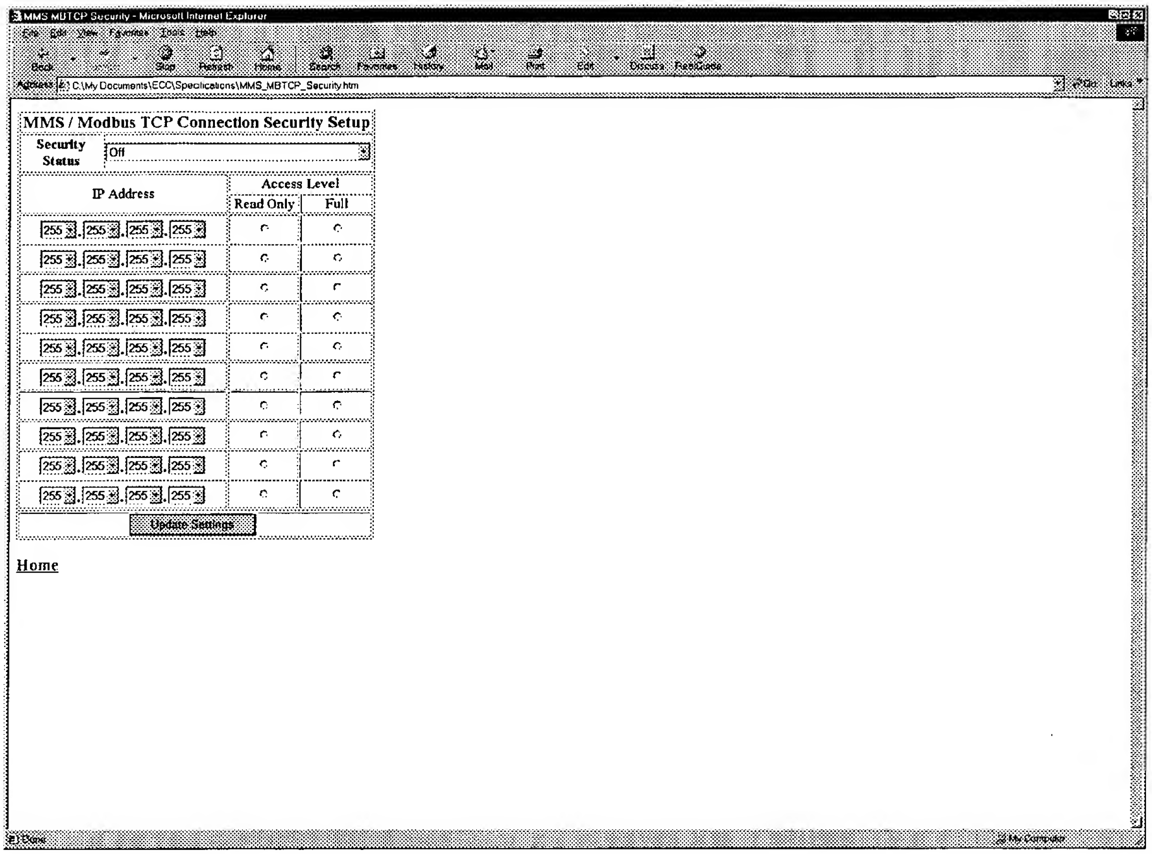


Figure 9 – MMS/ModbusTCP Security Setup View

REDACTED

3.1.3.7 HTML ECC Advanced Parameters Setup

The ECC will have an advanced parameters setup page that will be accessible by the administrator password only. This setup page will allow for advanced users to "tweak" ECC timing values that normally should never be changed. The administrator will also be able to change the default language the ECC will use at the login page. The following table shows the storage format for these values.

Register	HiByte	LoByte
507	ModbusTCP Client Timeout in seconds (5 – 60)	MMSTCP Client Timeout in seconds (5 – 60)
508	HTML Access token expiration time in minutes (1 – 255)	
509	RS485 Timeout in seconds (3 – 10)	DPR Timeout in seconds (3 – 10)
510	CM/RS485 Time Synchronization Interval in seconds (0 = disabled, 30 – 65535)	
511	HTML default language type (0 = English, 1 = French, 2 = Spanish)	

Table 24 – Advanced Setup Parameters

REDACTED

The following figure is what the MMS/ModbusTCP Security Setup HTML page will potentially look like. This page will be accessible by the administrator password only.

Advanced Setup

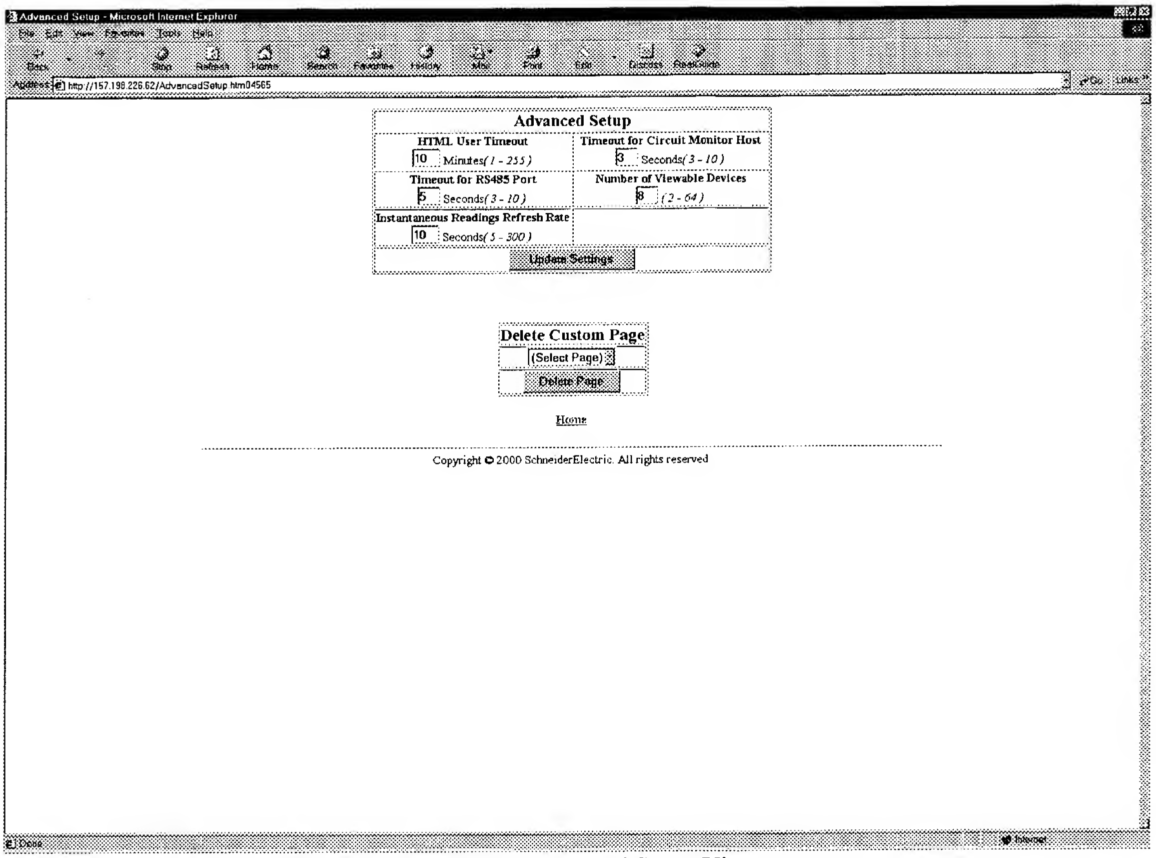


Figure 10 – Advanced Setup View

REDACTED

3.1.4 HTML Diagnostics View

The ECC will have the ability to accumulate diagnostics information for display to the user for troubleshooting or performance knowledge. Also, the ECC version information will be shown here. The counters and watermarks will be volatile and are shown in the following table.

Counters/Watermarks
RS485 Timeouts
RS485 Checksum Errors
RS485 Protocol Errors
RS485 Outbound Read Messages
RS485 Inbound Read Messages
RS485 Outbound Write Messages
RS485 Inbound Write Messages
Dual Port Ram Timeouts
Dual Port Ram Checksum Errors
Dual Port Ram Protocol Errors
Dual Port Ram Outbound Read Messages
Dual Port Ram Inbound Read Messages
Dual Port Ram Outbound Write Messages
Dual Port Ram Inbound Write Messages
MMS Timeouts (Future)
MMS Protocol Errors (Future)
MMS Outbound Read Messages (Future)
MMS Inbound Read Messages (Future)
MMS Outbound Write Messages (Future)
MMS Inbound Write Messages (Future)
MMS Inbound Connections (Future)
MMS Outbound Connections (Future)
MMS Active Inbound Connections (Future)
MMS Active Outbound Connections (Future)
MMS Maximum Inbound Connections (Future)
MMS Maximum Outbound Connections (Future)
ModbusTCP Timeouts
ModbusTCP Protocol Errors
ModbusTCP Outbound Read Messages
ModbusTCP Inbound Read Messages
ModbusTCP Outbound Write Messages
ModbusTCP Inbound Write Messages
ModbusTCP Inbound Connections
ModbusTCP Outbound Connections
ModbusTCP Active Inbound Connections
ModbusTCP Active Outbound Connections
ModbusTCP Maximum Inbound Connections
ModbusTCP Maximum Outbound Connections
Ethernet CRC Errors
Ethernet Alignment Errors
Ethernet Code Errors
Ethernet Long Frame Errors
Ethernet Short/Runt Frame Errors
Ethernet Maximum Collision

Table 25 – ECC Diagnostic Counters/Watermarks

REDACTED

The following figure is what the HTML Diagnostics View page will potentially look like. The "Reset" button will not be seen if the user has "view only" access.

Diagnostics View

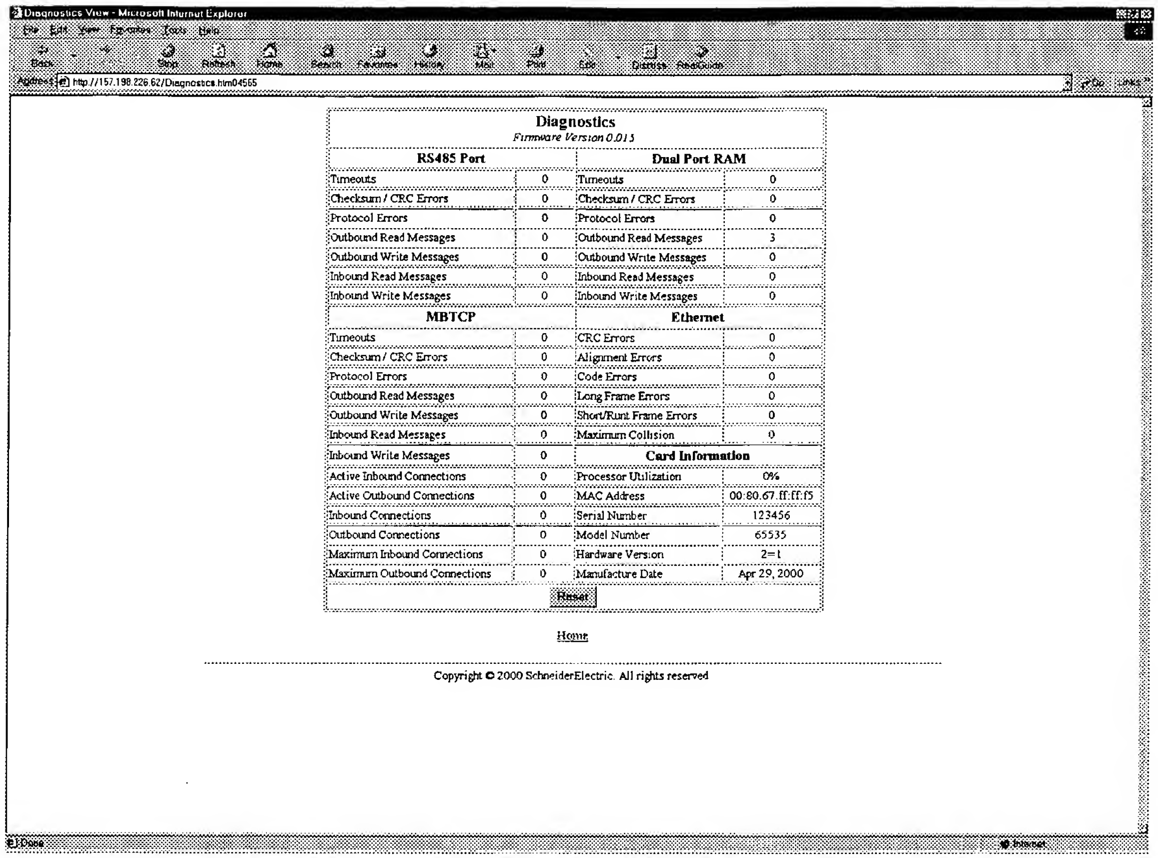


Figure 11 – Diagnostics View

3.1.5 Multilingual HTML (Future)

To be able to have the ECC available outside the United States and even more acceptable within, it will need to have all displays available in English, French, and Spanish. The user may choose a language for the browser session at login. The default language selection at login will be English, but the default language may be changed by entering the Advanced Setup HTML page with the administrator password. This means that all parts of the ECC firmware that has viewable strings will have to be designed in such a fashion to be displayed in multiple languages. The following table shows the storage of the HTML language type.

Register	HiByte	LoByte
511	HTML default language type (0 = English, 1 = French, 2 = Spanish)	

Table 26 – HTML Default Language

REDACTED

The following figure is what the Login HTML page will potentially look like pertaining to being Multilingual.

Log In

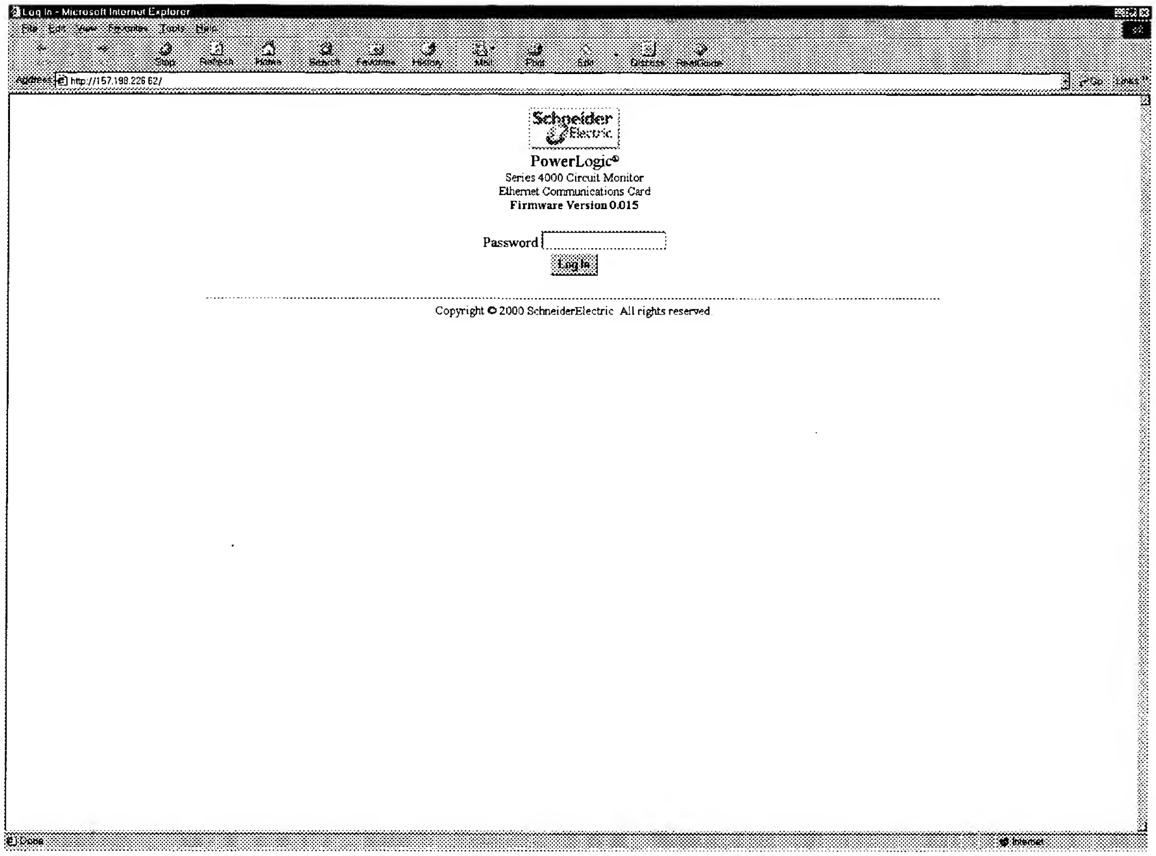


Figure 12 – Login View

3.1.6 HTML Home Page

The user will be transferred to the ECC home page after the login. Only the links that the user has access to will appear in the list. The following figure is what the ECC home page will potentially look like.

Home Page

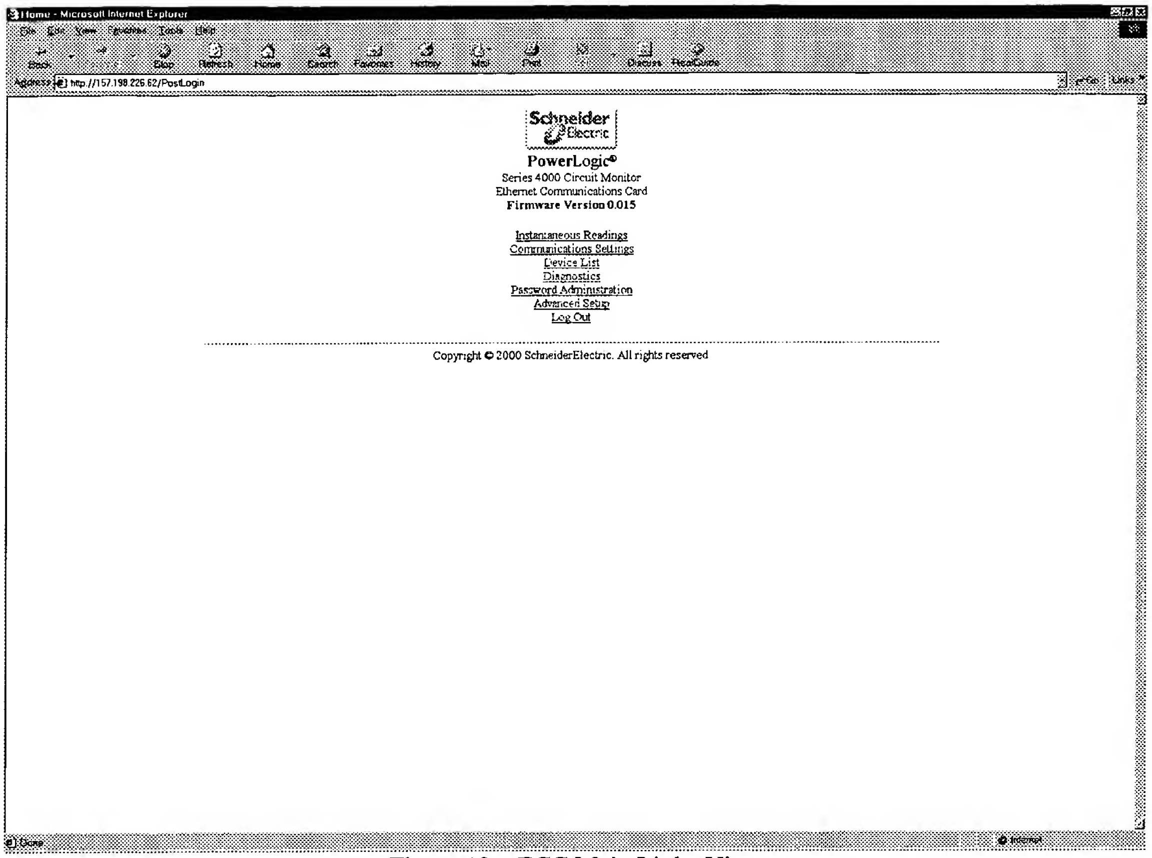


Figure 13 – ECC Main Links View

3.1.7 HTML Real Time Data Device Tables

The ECC will have the capability to show limited, real time, tabular data from the attached devices to the users in the form of HTML pages. These pages will consist of one static page for viewing information from the CM the ECC is inserted into and the possibility for up to five custom/downloadable HTML pages for viewing devices on the local RS485 daisy chain. These pages will be stored in the CM that the ECC is inserted into and must not exceed 20 kilobytes in size each.

REDACTED

3.1.7.1 HTML CM Instantaneous Readings View

The following shows the possible look of the CM device HTML real-time data table. This page will be configurable by the administrator password for view only or no access.

CM Instantaneous Readings View

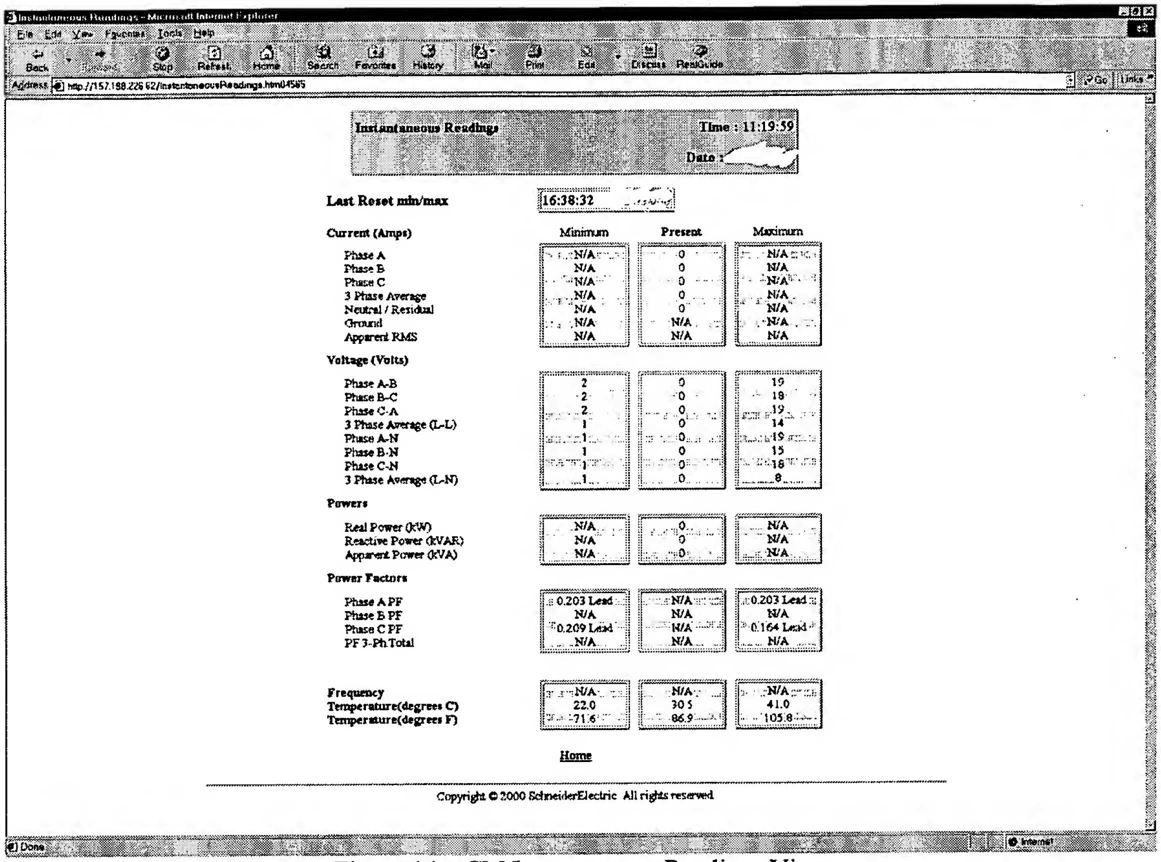


Figure 14 – CM Instantaneous Readings View

3.1.7.2 HTML Custom Device Readings Tables (Future)

The custom device tables will be available for use preferably by our Engineering Services group or by users very knowledgeable in the realm of HTML and JavaScript. The pages will be written in HTML with special delimiters that will tell the ECC to dynamically get register information from a device. Great care will have to be taken by the page author to try to get the optimal performance out of these pages and to keep the sizes within the allowed range. The delimiters at the beginning (PL__) and end (__PL) of a string signify to the ECC to parse this string and dynamically fill it with register data. The following is a table of the supported PowerLogic tags and also, an HTML example of how they could be used.

REDACTED

Function Code	Function Name	PowerLogic Tag
0	SyMax Block Read - Registers	<DeviceID>^<RegisterAddress>[<NumberOfRegisters>] example tag = PL__1^1003[5]__PL example of data returned = 85,86,84,25,56
4	SyMax Scattered Read – Registers	<DeviceID>^<RegisterAddress1>,<RegisterAddress2>,etc example tag = PL__1^1003,1004,1005,1006,1007__PL example of data returned = 85,86,84,25,56
3	Modbus Block Read – Holding Registers	<DeviceID>^H<RegisterAddress>[<NumberOfRegisters>] example tag = PL__1^H1003[5]__PL example of data returned = 85,86,84,25,56
4	Modbus Block Read – Input Registers	<DeviceID>^I<RegisterAddress>[<NumberOfRegisters>] example tag = PL__1^I1003[5]__PL example of data returned = 85,86,84,25,56
100	Modbus Scattered Read – Holding Registers	<DeviceID>^S<RegisterAddress1>,<RegisterAddress2>,etc example tag = PL__1^S1003,1004,1005,1006,1007__PL example of data returned = 85,86,84,25,56

Table 27 – PowerLogic HTML Tags

REDACTED

Source:

```
<html>

<head>
<META HTTP-EQUIV="refresh" CONTENT="5">
<title>CM2350 - Slave Device 3</title>
</head>

<body>
<form name="view_form">
    <p align="center">
      <input type = "text" name = "time_spot" size = "40">
      <table border="1" width="600">
       <tr>
           <td width="600"><p align="center"><font size="4"><b>CM2350 - Slave
             Device 3</b></font></p>
           </td>
       </tr>
    </table>
    <table border="1" width="600">
        <tr>
           <td width="300">
             <p align="center">Frequency</p>
           </td>
           <td align="center" width="90"><p align="center"><input
           type="text" size="5" name="frequency"></p>
           <td width="100">
             <p align="center">Hz</p>
           </td>
        </tr>
        <tr>
           <td width="300">
             <p align="center">Current Phase A</p>
           </td>
           <td align="center" width="90"><p align="center"><input
           type="text" size="5" name="currentphasea"></p>
           <td width="100">
             <p align="center">Amps</p>
           </td>
        </tr>
        <tr>
           <td width="300">
             <p align="center">Current Neutral</p>
           </td>
           <td align="center" width="90"><p align="center"><input
           type="text" size="5" name="currentneutral"></p>
           <td width="100">
             <p align="center">Amps</p>
           </td>
        </tr>
        <tr>
           <td width="300">
             <p align="center">Current Ground</p>
           </td>
           <td align="center" width="90"><p align="center"><input
           type="text" size="5" name="currentground"></p>
           <td width="100">
             <p align="center">Amps</p>
           </td>
        </tr>
    </table>
```

REDACTED

```
<br><HR SIZE="1" width="66%"><CENTER><font face="Times Roman"
size="2">Copyright ©     SchneiderElectric. All rights
reserved.</font></CENTER>
</form>

<script language="JavaScript">
function ShowFreq()
{
    Registers = [PL__3^2020,2021,2022,2025,1001,1003,1006,1007__PL];
    ScaleFactorA = Registers[0];
    ScaleFactorB = Registers[1];
    ScaleFactorC = Registers[2];
    ScaleFactorF = Registers[3];
    Frequency = Registers[4];
    CurrentPhaseA = Registers[5];
    CurrentNeutral = Registers[6];
    CurrentGround = Registers[7];
    ScaleFactorAMultiplier = 0;
    ScaleFactorBMultiplier = 0;
    ScaleFactorCMultiplier = 0;
    ScaleFactorFMultiplier = 0;
    TheTime = new Date();

    switch (ScaleFactorA)
    {
        case -2:
            ScaleFactorAMultiplier = 0.01;
            break;
        case -1:
            ScaleFactorAMultiplier = 0.1;
            break;
        case 1:
            ScaleFactorAMultiplier = 10;
            break;
        default:
            ScaleFactorAMultiplier = 1;
            break;
    }
    switch (ScaleFactorB)
    {
        case -2:
            ScaleFactorBMultiplier = 0.01;
            break;
        case -1:
            ScaleFactorBMultiplier = 0.1;
            break;
        case 1:
            ScaleFactorBMultiplier = 10;
            break;
        default:
            ScaleFactorBMultiplier = 1;
            break;
    }
    switch (ScaleFactorC)
    {
        case -2:
            ScaleFactorCMultiplier = 0.01;
            break;
        case -1:
            ScaleFactorCMultiplier = 0.1;
            break;
        case 1:
            ScaleFactorCMultiplier = 10;
```

REDACTED

```
            break;
        default:
            ScaleFactorCMultiplier = 1;
            break;
    }
    switch (ScaleFactorF)
    {
        case -1:
            ScaleFactorFMultiplier = 0.1;
            break;
        default:
            ScaleFactorFMultiplier = 0.01;
            break;
    }
    Frequency *= ScaleFactorFMultiplier;
    CurrentPhaseA *= ScaleFactorAMultiplier;
    if (CurrentNeutral == -32768)
        CurrentNeutral = "N/A";
    else
        CurrentNeutral *= ScaleFactorBMultiplier;
    if (CurrentGround == -32768)
        CurrentGround = "N/A";
    else
        CurrentGround *= ScaleFactorCMultiplier;
    document.view_form.frequency.value = Frequency;
    document.view_form.currentphasea.value = CurrentPhaseA;
    document.view_form.currentneutral.value = CurrentNeutral;
    document.view_form.currentground.value = CurrentGround;
    document.view_form.time_spot.value = TheTime;
}
ShowFreq();
</script>

</body>

</html>
```

REDACTED

View:

Proof

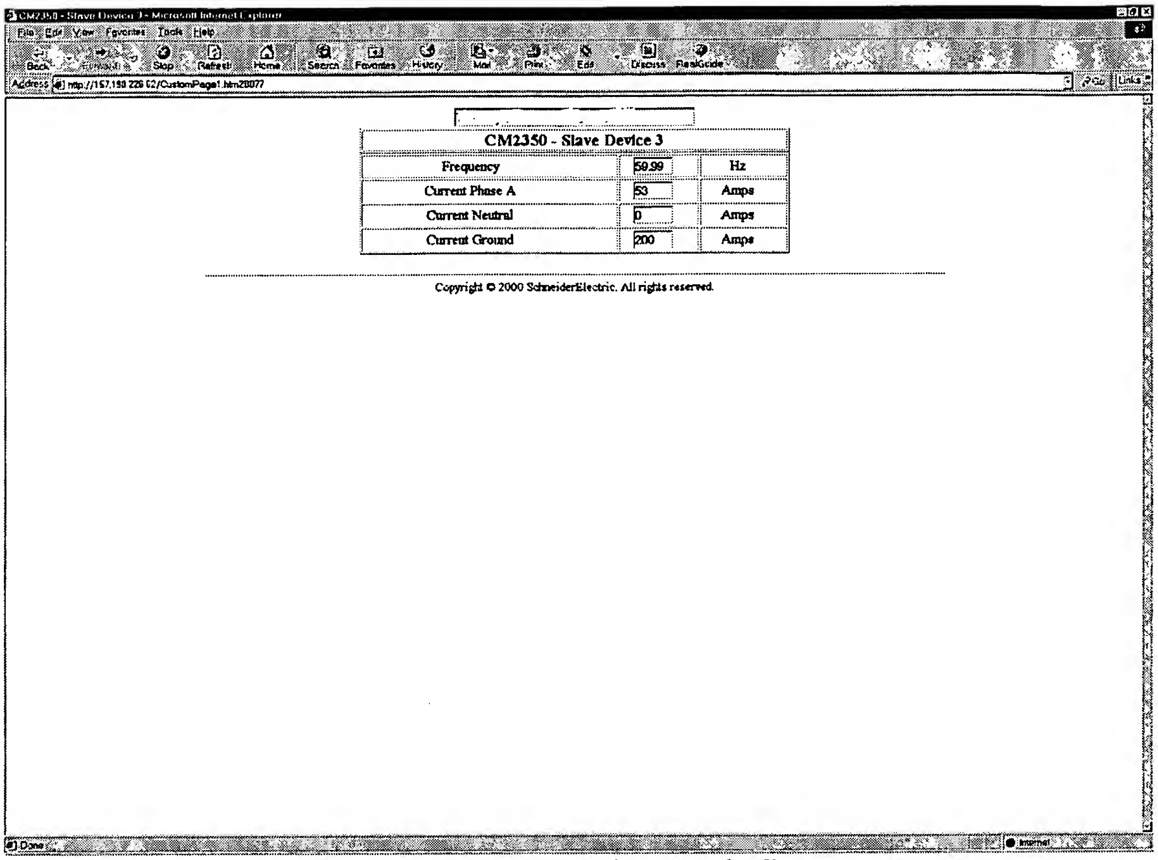


Figure 15 – Custom Device Example View

REDACTED

3.2 ModbusTCP Server

The ECC will be able to provide the ModbusTCP server functionality to allow external ModbusTCP clients access to the ECC's attached slave devices and the CM the ECC is inserted into. The ECC will post TCP/IP connection listens on TCP port 502. The ECC will allow a maximum of ten ModbusTCP clients to connect at any given time.

3.2.1 ModbusTCP Server Function Codes

The following table shows the supported ModbusTCP function codes the ECC will allow.

Function Code	Sub-function Code	Description	Availability
1	X	Block Read - Coil Status	Phase I
2	X	Block Read - Input Coil Status	Phase I
3	X	Block Read - Holding Registers	Phase I
4	X	Block Read - Input Registers	Phase I
5	X	Single Write - Coil	Phase I
6	X	Single Write - Holding Register	Phase I
16	X	Block Write - Holding Registers	Phase I
20	X	Block Read - General Reference/File	Phase I
21	X	Block Write - General Reference/File	Phase I
100	4	Scattered Read - Holding Registers	Phase I
101	50	Set Time	Future
102	X	Security Operations	Phase I

Table 28 – ModbusTCP Server Function Codes

3.3 MMSTCP Server (Future)

The ECC will be able to provide the MMSTCP (RFC-1006) server functionality to allow external PowerLogic compatible MMSTCP clients access to the ECC's attached slave devices. The ECC will allow a maximum of ten MMSTCP connections at any given time. The ECC will post MMSTCP connection listens on TCP port 102 with the MMS parameters in the following table.

MMS Parameter	Value
AP Title	0 0 00 0
AP Invoke ID	0
AE Qualifier	0
AE Invoke ID	0
Psel	00 00 00 02
Ssel	00 01
Tsel	00 01

Table 29 – MMSTCP Server Connect Parameters

REDACTED

3.3.1 MMSTCP Services (Future)

The following table shows the MMS services supported in the ECC and gives a slight definition of how each is used.

MMS Service	Description
Initiate	The ECC responds to all Initiate indications and is able to initiate connections for sub-net initiated communications.
Conclude	The ECC is able to respond to a Conclude indication and is able to conclude a connection in the case of inactivity on a channel or a request from sub-net initiated communications.
Abort	The ECC can respond to an Abort indication and generate a serial error code, if needed, in the case of sub-net initiated communications.
Cancel	The ECC responds to a Cancel by generating a serial error code, if needed, in the case of sub-net initiated communications.
Reject	The ECC can receive a Reject indication from another device and generate a serial error code, if needed, in the case of sub-net initiated communications.
Identify	The ECC returns the appropriate device information (vendor name, model, and firmware version) in response to an Identify indication.
Variable Read/Write	The ECC responds to data access requests for symbolically addressed variables. Additionally, the ECC accepts serial read/write requests from a sub-net initiating device, converts them to a MMS symbolic address request, and returns the data appropriately.

Table 30 – ECC Supported MMSTCP Services

3.3.2 MMSTCP Server Security Backwards Compatibility (Future)

Due to the new IP connection-based security implementation, the ECC will not need to support the old security scheme implemented in existing EGWs. This should not cause any problems in systems still needing security or even in existing systems where an ECC is installed with other EGWs. The new mechanism in the ECC can still be utilized in conjunction with the old mechanism in the EGWs with no problems.

3.3.3 MMSTCP Symbolic Variable Strings (Future)

Character-based symbolic addressing will be used for data access. The basic format will be a two-part address separated by a "^". The first part is the end Device ID; the second part is protocol-specific. The following table shows the representation of each of the MMSTCP symbolic variable representations of the sub-network requests.

SyMax Function Code	Name	SyMax Format for MMS
0	Block Read - Registers	<DeviceID>^<RegisterAddress>[<NumberOfRegisters>], *,
2	Block Write - Registers	<DeviceID>^<RegisterAddress>[<NumberOfRegisters>], *,
4	Scattered Read - Registers	<DeviceID>^<RegisterAddress>, <DeviceID>^<RegisterAddress>,
4	Record Read - File	<DeviceID>^<RegisterAddress>[<NumberOfRegisters>], <DeviceID>^RecordNumber[r]

Table 31 – MMSTCP SyMax Symbolic Variable Formats

REDACTED

Modbus /Jbus Function Code	Name	Modbus/Jbus Format for MMS
1	Block Read – Coil Status	<DeviceID>^C<CoilAddress>[<NumberOfCoils>], *,
2	Block Read – Input Coil Status	<DeviceID>^D<CoilAddress>[<NumberOfCoils>], *, *
3	Block Read – Holding Registers	<DeviceID>^H<RegisterAddress>[<NumberOfRegisters>], *, *
4	Block Read – Input Registers	<DeviceID>^I<RegisterAddress>[<NumberOfRegisters>] *,
5	Single Write – Coil	<DeviceID>^C<CoilAddress>
6	Single Write – Register	<DeviceID>^H<RegisterAddress>
16	Block Write – Holding Registers	<DeviceID>^H<RegisterAddress>[<NumberOfRegisters>], *, *
20	Block Read – General Reference/File	<DeviceID>^F<RegisterAddress>[<NumberOfRegisters>]<FileNumber>, *, *
21	Block Write - General Reference/File	<DeviceID>^F<RegisterAddress>[<NumberOfRegisters>]<FileNumber>, *, *
100	Scattered Read – Holding Registers	<DeviceID>^S<RegisterAddress>, <DeviceID>^S<RegisterAddress>, <DeviceID>^S<RegisterAddress>
102	Security Operations	<DeviceID>^Z<SecurityCode>^ regular formats minus device ID

Table 32 – MMSTCP Modbus/Jbus Symbolic Variable Formats

3.3.3.1 MMSTCP Symbolic Variable String Examples (Future)

A SyMax Block Read of registers 1002 through 1004 from a SyMax device defined as Device ID 12 in the ECC should be done with a single MMS read of a list of 3 individual symbolic variables in the following format:

12^1002[3]	= Variable 1
*	= Variable 2
*	= Variable 3

A SyMax Block Write of registers 6800 through 6802 from a SyMax device defined as Device ID 12 in the ECC should be done with a single MMS write of a list of 4 individual symbolic variables in the following format:

12^6800[3]	= Variable 1	-32768 to 32767 for data
*	= Variable 2	-32768 to 32767 for data
*	= Variable 3	-32768 to 32767 for data
*	= Variable 4 with mask value to be applied to the written registers' data	

Note that there is (1 + NumberOfRegisters) variables. This is because the last variable's data must be used as a mask value in the SyMax operation. This last variable's data is usually FFFF because all the bits for

REDACTED

all the values are to be written into the associated registers. The FFFF value can be changed if only certain bits are to be written.

A SyMax Scattered Read of registers 1002, 1004, 1008 from a SyMax device defined as Device ID 12 in the ECC should be done with a single MMS read of a list of 3 individual symbolic variables in the following format:

12^1002	= Variable 1
12^1004	= Variable 2
12^1008	= Variable 3

A Modbus/Jbus Block Read of coils 20 through 22 from a Modbus/Jbus device defined as Device ID 12 in the ECC should be done with a single MMS read of a list of 3 individual symbolic variables in the following format:

12^C20[3]	= Variable 1
*	= Variable 2
*	= Variable 3

A Modbus Block Read of Input coils 100020 through 100022 from a Modbus device defined as Device ID 12 in the ECC should be done with a single MMS read of a list of 3 individual symbolic variables in the following format:

12^D20[3]	= Variable 1
*	= Variable 2
*	= Variable 3

A Jbus Block Read of Input coils 20 through 22 from a Jbus device defined as Device ID 12 in the ECC should be done with a single MMS read of a list of 3 individual symbolic variables in the following format:

12^D20[3]	= Variable 1
*	= Variable 2
*	= Variable 3

A Modbus Block Read of Holding registers 401002 through 401004 from a Modbus device defined as Device ID 12 in the ECC should be done with a single MMS read of a list of 3 individual symbolic variables in the following format:

12^H1002[3]	= Variable 1
*	= Variable 2
*	= Variable 3

A Jbus Block Read of Holding registers 1002 through 1004 from a Jbus device defined as Device ID 12 in the ECC should be done with a single MMS read of a list of 3 individual symbolic variables in the following format:

12^H1002[3]	= Variable 1
*	= Variable 2
*	= Variable 3

A Modbus Block Read of Input registers 301002 through 301004 from a Modbus device defined as Device ID 12 in the ECC should be done with a single MMS read of a list of 3 individual symbolic variables in the following format:

12^I1002[3]	= Variable 1
*	= Variable 2
*	= Variable 3

A Jbus Block Read of Input registers 1002 through 1004 from a Jbus device defined as Device ID 12 in the ECC should be done with a single MMS read of a list of 3 individual symbolic variables in the following format:

12^I1002[3]	= Variable 1
*	= Variable 2

REDACTED

*	= Variable 3	

A Modbus/Jbus Single Coil Write to coil 20 to a Modbus/Jbus device defined as Device ID 12 in the ECC should be done with a single MMS write of a single symbolic variable in the following format:

12^C20	= Variable 1	1 or 0 for data

A Modbus Single Register Write to register 406800 to a Modbus device defined as Device ID 12 in the ECC should be done with a single MMS write of a single symbolic variable in the following format:

12^H6800	= Variable 1	-32768 to 32767 for data

A Jbus Single Register Write to register 6800 to a Jbus device defined as Device ID 12 in the ECC should be done with a single MMS write of a single symbolic variable in the following format:

12^H6800	= Variable 1	-32768 to 32767 for data

A Modbus Block Write to registers 406800 to 406802 to a Modbus device defined as Device ID 12 in the ECC should be done with a single MMS write of a list of 3 individual symbolic variables in the following format:

12^H6800[3]	= Variable 1	-32768 to 32767 for data
*	= Variable 2	-32768 to 32767 for data
*	= Variable 3	-32768 to 32767 for data

A Jbus Block Write to registers 6800 to 6802 to a Jbus device defined as Device ID 12 in the ECC should be done with a single MMS write of a list of 3 individual symbolic variables in the following format:

12^H6800[3]	= Variable 1	-32768 to 32767 for data
*	= Variable 2	-32768 to 32767 for data
*	= Variable 3	-32768 to 32767 for data

A Modbus/Jbus Read from General Reference File 1 registers 2 to 4 from a Modbus/Jbus device defined as Device ID 12 in the ECC should be done with a single MMS read of a list of 3 individual symbolic variables in the following format:

12^F2[3]1	= Variable 1
*	= Variable 2
*	= Variable 3

A Modbus/Jbus Write to General Reference File 1 registers 2 to 4 to a Modbus/Jbus device defined as Device ID 12 in the ECC should be done with a single MMS write of a list of 3 individual symbolic variables in the following format:

12^F2[3]1	= Variable 1	-32768 to 32767 for data
*	= Variable 2	-32768 to 32767 for data
*	= Variable 3	-32768 to 32767 for data

A Modbus Scattered Read of registers 401002, 401004, 401008 from a Modbus device defined as Device ID 12 in the ECC should be done with a single MMS read of a list of 3 individual symbolic variables in the following format:

12^S1002	= Variable 1
12^S1004	= Variable 2
12^S1008	= Variable 3

A Jbus Scattered Read of registers 1002, 1004, 1008 from a Jbus device defined as Device ID 12 in the ECC should be done with a single MMS read of a list of 3 individual symbolic variables in the following format:

12^S1002	= Variable 1
12^S1004	= Variable 2
12^S1008	= Variable 3

REDACTED

A Modbus Secured Scattered Read of registers 401002, 401004, 401008 from a Modbus device defined as Device ID 12 in the ECC should be done with a single MMS read of a list of 3 individual symbolic variables in the following format:

12^Z3546^S1002	=Variable 1
12^Z3546^S1004	= Variable 2
12^Z3546^S1008	= Variable 3

3.4 ModbusTCP Client (Future)

The ECC will have the ability to initiate communications as a ModbusTCP client when configured to do so based on the Remote Device Connections Setup. The ECC will do all transactions based on a connection to TCP port 502 and will support the same Modbus function codes listed in the ModbusTCP Server Function Codes Section. Each read or write operation will be accompanied by a connect and a disconnect message over the Ethernet.

3.5 MMSTCP Client (Future)

The ECC will have the ability to initiate communications as a MMSTCP client when configured to do so based on the Remote Device Connections Setup. The ECC will do all transactions based on a connection to TCP port 102 and will support the MMS functionality explained in the MMSTCP Services Section. Each read or write operation will be accompanied by a connect and a disconnect message over the Ethernet.

3.6 SNTP Server (Future)

The ECC will have the ability to listen on TCP port 123 for an SNTP server to send time synchronization information to the ECC. This information will then update the ECC's internal time and will allow this updated internal time to be used in the time synchronization mechanism which will update all the attached devices (CM and RS485) to the ECC on configurable intervals.

REDACTED

3.7 FTP Server

The ECC has the ability to act as an FTP server. This capability will allow the ECC to listen on the Ethernet at TCP port 21 for file transfers. Utilizing this capability will allow a good mechanism for quick downloadable ECC basic setup, downloadable ECC firmware updates, and downloadable custom device HTML tables.

3.7.1 FTP Download for Basic Setup (Future)

The ECC will have the ability to accept an FTP download to accomplish limited setup changes/additions. The initial setup of the ECC through the CM display must already be completed to get the ECC configured properly on the TCP/IP network prior to attempting the setup download. Most any desktop PC running Windows NT/95/98 and using TCP/IP will have the capability to FTP a setup file to the ECC. The setup file will be an ASCII text file with a list of strings for the ECC to execute. The user will have to open an FTP session from within the DOS prompt and "put/send" the text file to the ECC. The text file must be named "setup.txt" and the user must login to the ECC FTP server with the administrator password (user name is ignored by the ECC). The following table shows what each line in the text file should be.

Line Number	Description
1	RS485 Port Baud Rate (1200, 2400, 4800, 9600, 19200, 38400)
2	RS485 Port Parity (0 = none, 2 = even)
3	RS485 Port Timeout in seconds (3 – 10)
4	RS485 Time Sync Interval in seconds (0 = disabled, 30 – 65535)
5	Device ID 2's RS485 Address (0 – 247)
6	Device ID 2's RS485 Protocol (4 = PowerLogic, 6 = Modbus, 11 = Jbus, any value other than these specified will remove this device from the list)
7	Device ID 3's RS485 Address (0 – 247)
8	Device ID 3's RS485 Protocol (4 = PowerLogic, 6 = Modbus, 11 = Jbus, any value other than these specified will remove this device from the list)
.	
66	Through Device ID 32's info

Table 33 – Basic Setup FTP File Format

The following shows what an example setup text file would look like.

9600
2
5
300
3
4
5
6
10
11
24
4

Note: A "Wizard" application could be provided here to automatically generate this file for the user.

REDACTED

3.7.2 FTP Download for Firmware Update

The ECC will have the ability to accept an FTP transfer to accomplish a firmware download. The ECC must be configured properly on the TCP/IP network prior to attempting the firmware download. Most any desktop PC running Windows NT/95/98 and using TCP/IP will have the capability to FTP the firmware file to the ECC. The user will have to open an FTP session from within the DOS prompt and "put/send" the firmware file to the ECC. The file will be named "firmware.bin" and the user must login to the ECC FTP server with the administrator password (user name is ignored by the ECC). The new firmware binary will be accepted by the ECC into DRAM space. Once the FTP transaction is complete, the firmware binary will be copied into flash, and the ECC will reset to load and run with the new image.

3.7.3 FTP Download for Custom Device View Tables (Future)

The ECC will have the ability to accept an FTP download to allow for five custom designed HTML device view pages. The ECC must be configured properly on the TCP/IP network prior to attempting the download. Most any desktop PC running Windows NT/95/98 and using TCP/IP will have the capability to FTP the file to the ECC. The HTML file will be an ASCII text file written in standard HTML language.

Note that it is up to the user to correctly construct the HTML page. The ECC will only confirm that there is a correct HTML title tag, and then accept the whole file as correct. Once the file is accepted, a link to the page will appear on the ECC Home page. The link name will be the page's title up to 32 characters.

The user will have to open an FTP session from within the DOS prompt and "put/send" the HTML file to the ECC. The user must login to the ECC FTP server with the administrator password (user name is ignored by the ECC). Once the file is accepted by the ECC, the HTML file's title will show up in the list on the ECC Home page.

3.7.3.1 Post FTP Download for Custom Device View Tables (Future)

The ECC will normally have the custom HTML pages loaded into DRAM at runtime. When a new one is downloaded, the ECC will update its image in DRAM then have to store the HTML page in a file in the CM. According to the CM Design Specification, the following table signifies each file for the pages.

File Type	Description	File Size	Record Size
35	HTML File #1	Dynamic	Static
36	HTML File #2	Dynamic	Static
37	HTML File #3	Dynamic	Static
38	HTML File #4	Dynamic	Static
39	HTML File #5	Dynamic	Static

Table 34 – Downloadable HTML Page Storage

Each downloadable HTML page will have to be buffered to fit the static record size. This means that the HTML page will have to be evenly divisible by 100 because each record size is limited to 100bytes.

4 Serial Communications

The ECC will have the ability to have its single RS485 port utilized for communications in combinations of the following: 2-wire, 4-wire, multi-protocol, master, slave (Future). To try to keep with the goal of having the user do as little setup as possible, the goal of the ECC RS485 logistics will be to have only two modes of operation. These two modes will be able to be as "dynamic" as possible and respond/react to the protocols supported as needed. These two modes will be called "4-wire Smart Mode" and "2-wire Smart Mode".

REDACTED

4.1 4-wire Smart Mode

The 4-wire Smart Mode will be the default and primary setup for the ECC RS485 communications. This mode will allow for full-duplex bi-directional communications when needed. The ECC will be able to "know" when the port is being utilized as a slave or master by reacting to what is connected to it during communications. This "knowledge" also includes the ability to determine between SyMax, Modbus, and Jbus protocols on the fly. This ability helps the ECC adept even better to the use of "mixed-protocol" devices.

4.2 2-wire Smart Mode

The 2-wire Smart Mode will be the secondary setup for the ECC RS485 communications. This mode will allow for half-duplex communications only. This means that all the devices (master and slaves) together share the communication signals and can only communicate one at a time. In this mode, just like the 4-wire Smart Mode, the ECC will be able to "know" when the port is being utilized as a slave or master by reacting to what is connected to it during communications. This mode will accommodate Modbus/Jbus 2-wire communications only. This ability also helps ease the ECC setup for the user.

5 Inter-processor Communications

The primary purpose of the ECC is for high-speed communications to the CM the ECC is inserted into. This inter-processor communications will be based on a high-speed dual-port RAM interface between the bus of the ECC's processor and the bus of the CM's processor. The actual speed of the communications will be determined during development/testing.

5.1 Inter-processor Communications Theory of Operation

The inter-processor communications will be based on a proprietary "flavor" of Modbus. This choice was made since the Modbus protocol is one already well adapted into both devices, and since the characteristics of the protocol fit the dual-port RAM interface fairly well. Because the CM will have the ability to initiate communications and cause a potential for full-duplex, bi-directional communications (collisions) through the dual-port RAM in the ECC, the dual-port RAM will have to be split into four main memory regions.

One half (upper half), from the point-of-view of the ECC, will be the incoming side for communications from the CM. This half will then be split into two equal memory regions that will be designated for incoming commands and incoming replies. The other half (lower half), again from the point-of-view of the ECC, will be the outgoing side for communications to the CM. This half will also be split into two more equal memory regions that will be designated for outgoing commands and outgoing replies.

To ease the message management on the CM and the ECC, no more than one command message per direction should be outstanding at any given time. There may also need to be "control" messages between just the CM and the ECC for movement of specific types of data unknown to the "outside". In these cases, the CM and the ECC should use a Modbus slave ID of 255 in the Modbus command message to signify to the device receiving the message that it is a "control" message.

As previously mentioned, there will be a slight adaptation of the Modbus packet structure. There will be two bytes (one word) added as the first part of the message that will be the count of the bytes contained in the rest of the message. The rest of the message will then be a regularly formatted Modbus message. The ECC will read the CM's RS485 slave port address out of the CM at boot and that address will be used for MBTCP communications to the CM the ECC is inserted into.

After the ECC fills the appropriate region of the dual-port RAM with the appropriate message needed, the offset address of 0x7FF should be written with the appropriate interrupt ID to trigger an interrupt to the CM processor. Likewise, after the CM fills the appropriate region of the dual-port RAM with the appropriate

REDACTED

message needed, the offset address of 0x7FE should be written with the appropriate interrupt ID to trigger an interrupt to the ECC processor.

The following diagram shows the memory map of the dual-port ram followed by an explanation of each of the numbered memory regions.

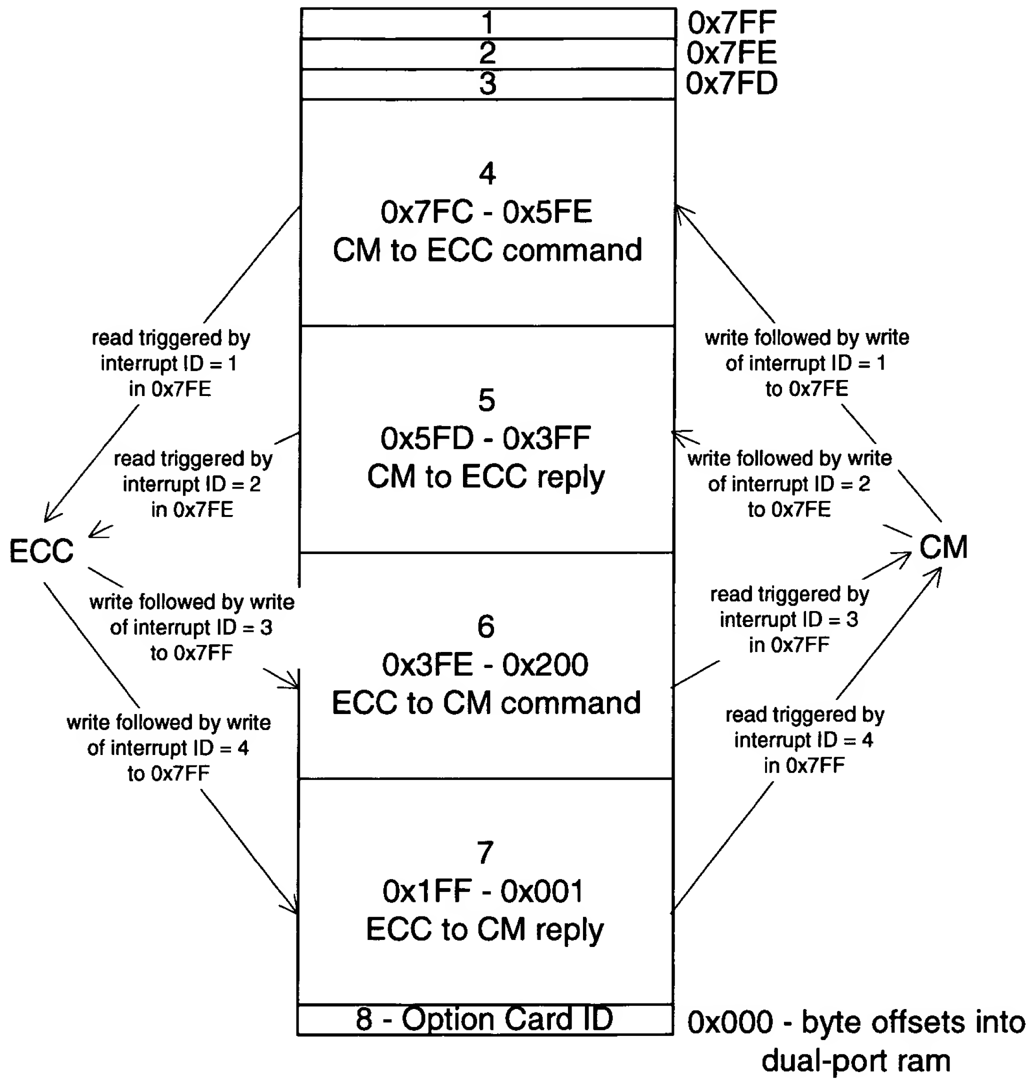


Figure 16 - Dual-Port Ram Memory Map

REDACTED

Memory Region	Description
1	The ECC writes the interrupt ID here to trigger an interrupt to the CM. Or The CM reads this address after being triggered by the interrupt to see what the interrupt ID is.
2	The CM writes the interrupt ID here to trigger an interrupt to the ECC. Or The ECC reads this address after being triggered by the interrupt to see what the interrupt ID is.
3	Reserved
4	The ECC, after interrupt and reading the interrupt ID from 0x7FE, will read this memory region from offset 0x600 for number of bytes shown in 0x5FE and 0x5FF to get a Modbus command message from the CM. Or The CM, before writing the appropriate interrupt ID value in 0x7FE, will write the Modbus command message in this memory region from offset 0x600 and write the number of bytes in the message into 0x5FE and 0x5FF.
5	The ECC, after interrupt and reading the interrupt ID from 0x7FE, will read this memory region from offset 0x401 for number of bytes shown in 0x3FF and 0x400 to get a Modbus reply message from the CM. Or The CM, before writing the appropriate interrupt ID value in 0x7FE, will write the Modbus reply message in this memory region from offset 0x401 and write the number of bytes in the message into 0x3FF and 0x400.
6	The CM, after interrupt and reading the interrupt ID from 0x7FF, will read this memory region from offset 0x202 for number of bytes shown in 0x200 and 0x201 to get a Modbus command message from the ECC. Or The ECC, before writing the appropriate interrupt ID value in 0x7FF, will write the Modbus command message in this memory region from offset 0x202 and write the number of bytes in the message into 0x200 and 0x201.
7	The CM, after interrupt and reading the interrupt ID from 0x7FF, will read this memory region from offset 0x003 for number of bytes shown in 0x001 and 0x002 to get a Modbus reply message from the ECC. Or The ECC, before writing the appropriate interrupt ID value in 0x7FF, will write the Modbus reply message in this memory region from offset 0x003 and write the number of bytes in the message into 0x001 and 0x002.
8	Option Card Identification

Table 35 - Dual-Port Ram Memory Region Descriptions

REDACTED

The following table shows the combinations for the flow of messages through the dual-port RAM and the associated interrupt ID.

Direction	Interrupt ID	Description
Incoming Command (memory region 4)	1 (command)	Incoming command from the CM to the ECC (designated by slave id 255), or from the CM to the ECC to be routed to the outside
Incoming Reply (memory region 5)	2 (reply)	Incoming reply from the CM to the ECC (designated by slave id 255), or from the CM to the ECC to be routed to the outside
Outgoing Command (memory region 6)	3 (command)	Outgoing command from the ECC to the CM (designated by slave id 255), or from the outside to be routed by the ECC to the CM
Outgoing Reply (memory region 7)	4 (reply)	Outgoing reply from the ECC to the CM (designated by slave id 255), or from the outside to be routed by the ECC to the CM.

Table 36 – Dual-port RAM Logistics

REDACTED

5.2 Inter-processor Communications Boot-up Sequence

Because the entire ECC configuration will be stored in the CM the ECC is inserted into, the ECC will have to be able to get this information from the CM at every boot-up. Also, if the user changes the initial setup information (IP Address, IP Subnet Mask, IP Router, or Ethernet Physical Connection) via the CM display, the CM will have to re-boot the ECC to have the new settings take effect.

At initial power-up, the ECC will set its CM option card identifier in the lowest address of the dual-port RAM as soon as possible before starting any of the main firmware tasks. At this point, the ECC will finish its boot sequence based on the last known good initial setup information. This information will be held in the first thirteen bytes of the EEPROM based on the following table.

Byte Offset into EEPROM	Description
0	IP Address – 1^{st} Octet
1	IP Address – 2^{nd} Octet
2	IP Address – 3^{rd} Octet
3	IP Address – 4^{th} Octet
4	IP Sub-net Mask – 1^{st} Octet
5	IP Sub-net Mask – 2^{nd} Octet
6	IP Sub-net Mask – 3^{rd} Octet
7	IP Sub-net Mask – 4^{th} Octet
8	IP Router Address – 1^{st} Octet
9	IP Router Address – 2^{nd} Octet
10	IP Router Address – 3^{rd} Octet
11	IP Router Address – 4^{th} Octet
12	Ethernet Physical Connection

Table 37 – Last Known Good Boot State

After the completion of the boot sequence based on the last known good information, the ECC will then begin the first of a sequence of Modbus reads getting the basic ECC setup information from the CM. If the ECC finds a difference between what it has as the last known good information and what the CM has as the initial setup information, the ECC will update its last known good information and request a reboot of the CM. Once there is a match between the initial setup information between the ECC and the CM, the ECC will finish its boot sequence with all the appropriate basic setup information and continue to request any advanced information (Custom HTML pages) from the CM.

5.3 CM Communications Card Identification

The CM option card identifier used by the ECC is defined by the CM and is the value of six according to the CM Design Specification. This number will be placed in the lowest offset of the dual-port RAM by the ECC as soon as possible during boot-up.

6 Manufacturing Parameters

The ECC will need some specific parameters set for each unit during manufacturing. Some of the parameters that appear to be needed at this time are the Organizationally Unique Identifier (OUI), the serial number, the model number, the hardware revision number, and the date of manufacture (DOM).

REDACTED

6.1 Organizationally Unique Identifier (OUI)

An OUI is a 24 bit globally unique assigned number referenced by various standards. An OUI is used in the family of 802 LAN standards (Ethernet, Token Ring, etc). The OUI defined in IEEE Std 802-1990 can be used to generate 48 bit Universal LAN MAC addresses to identify LAN and MAN stations uniquely, and Protocol Identifiers to identify public and private protocols. These are used in Local and Metropolitan Area Network applications. The relevant standards include CSMA/CD (IEEE Std 802.3, ISO 8802-3), Token Bus (IEEE Std 802.4, ISO 8802-4), Token Ring (IEEE Std 802.5, ISO/IEC 8802-5), IEEE Std 802.6 (ISO/IEC DIS 8802-6), and FDDI (ISO 9314-2).

The ECCs will carry Square D's IEEE OUI "signature" of 00-80-67. The Power Management Operations (PMO) has a range of the OUIs for Ethernet MAC addresses and will be 00-80-67-80-00-00 through 00-80-67-FF-FF-FF. This range gives PMO a total of 8,388,608 MAC addresses to distribute and be used among our Ethernet devices.

The unique MAC addresses will have to be able to be assigned to each individual ECC during manufacturing. There are a few possibilities to accomplish this. One possibility is to have the MAC assignment done by writing the EEPROM before board manufacture and/or have an EEPROM socket for removal of the EEPROM to have the ability to re-write if needed. The other option would be to have firmware capabilities to set/change the MAC address when needed. This option requires more functionality in the firmware and also leaves open the potential of "lost" or duplicate MAC addresses. The following table shows the storage format of the MAC address in the EEPROM on the ECC.

Byte Offset into EEPROM	Description
13	Byte one of the MAC address (most significant)
14	Byte two of the MAC address
15	Byte three of the MAC address
16	Byte four of the MAC address
17	Byte five of the MAC address
18	Byte six of the MAC address (least significant)

Table 38 – MAC Address Storage Map

6.2 Serial Number

Another piece of information that will need to be stored in the ECC EEPROM will be a serial number. Here again, there are a few possibilities to accomplish this. One possibility is to have the serial number assignment done by writing the EEPROM before board manufacture and/or have an EEPROM socket for removal of the EEPROM to have the ability to re-write if needed. Another option would be to have firmware capabilities to set/change the serial number when needed. A third option would be to come up with a scheme to derive a serial number from the MAC address. By having the MAC address set, the serial number would also inherently be set. The following table shows the storage format of the 32 bit value of the serial number in the EEPROM on the ECC.

Byte Offset into EEPROM	Description
19	Byte one of the Serial Number (most significant)
20	Byte two of the Serial Number
21	Byte three of the Serial Number
22	Byte four of the Serial Number (least significant)

Table 39 – Serial Number Storage Map

REDACTED

6.3 Model Number

Another piece of information that will need to be stored in the ECC EEPROM will be a model number. Here again, there are a few possibilities to accomplish this. One possibility is to have the model number assignment done by writing the EEPROM before board manufacture and/or have an EEPROM socket for removal of the EEPROM to have the ability to re-write if needed. The other option would be to have firmware capabilities to set/change the model number when needed. The following table shows the storage format of the 16 bit value of the model number in the EEPROM on the ECC.

Byte Offset into EEPROM	Description
23	Byte one of the Model Number (most significant)
24	Byte two of the Model Number (least significant)

Table 40 – Model Number Storage Map

6.4 Hardware Revision Number

Another piece of information that will need to be stored in the ECC EEPROM will be a hardware revision number. Here again, there are a few possibilities to accomplish this. One possibility is to have the hardware revision number assignment done by writing the EEPROM before board manufacture and/or have an EEPROM socket for removal of the EEPROM to have the ability to re-write if needed. The other option would be to have firmware capabilities to set/change the hardware revision number when needed. The following table shows the storage format of the four ASCII characters that represent the hardware revision number in the EEPROM on the ECC.

Byte Offset into EEPROM	Description
25	ASCII character one of the Hardware Revision Number
26	ASCII character two of the Hardware Revision Number
27	ASCII character three of the Hardware Revision Number
28	ASCII character four of the Hardware Revision Number

Table 41 – Model Number Storage Map

6.5 Date of Manufacture (DOM)

Another piece of information that will need to be stored in the ECC EEPROM will be a Date of Manufacture value. Here again, there are a few possibilities to accomplish this. One possibility is to have the date assignment done by writing the EEPROM before board manufacture and/or have an EEPROM socket for removal of the EEPROM to have the ability to re-write if needed. The other option would be to have firmware capabilities to set/change the date when needed. The following table shows the storage format of the date of manufacture in the EEPROM on the ECC.

Byte Offset into EEPROM	Description
29	Month byte of the DOM
30	Day byte of the DOM
31	Year byte of the DOM, since 1900
32	Hour byte of the DOM
33	Minute byte of the DOM
34	Seconds byte of the DOM

Table 42 – DOM Storage Map

6.6 Manufacture Process Info

REDACTED

The final piece of information that will need to be stored in the ECC EEPROM will be an area of memory that will be used as a "scratch pad" for use during Manufacturing. This area will be written during manufacturing/testing of each unit. The following table shows the region in the EEPROM dedicates for this purpose.

Byte Offset into EEPROM	Description
35 – 74	Manufacturing "scratch pad"

Table 43 – Manufacturing Process Info Storage Map

7 Environmental

The ECC design will attempt to meet the same environmental requirements as the CM. This will prove difficult, however, due to the lack of availability of Industrial Ethernet components capable of doing the needed speeds/interfaces. The following table will show acceptable ranges for the environmental requirements that the ECC will attempt to meet.

Description	Values
Operating Temperature Range	-40°C to +70°C
Storage Temperature Range	-40°C to +85°C
Humidity Rating	5% to 95% rh (non-condensing)
Vibration	Equivalent to CM

Table 44 – Environmental Requirements

8 Agency Compliance

The ECC design will attempt to meet the same agency compliance as the CM. The following table shows the agency compliance that the ECC will attempt to meet.

Type		Description
Electromagnetic Interference	Radiated Emissions	FCC Part 15 Class A/CE Heavy Industrial
Electromagnetic Interference	Conducted Emissions	FCC Part 15 Class A/CE Heavy Industrial
Electrostatic Discharge	Air Discharge	IEC pub 1000-4-2 level 4
Immunity to Electrical Fast Transients	Transients	ANSI/IEEE C37.90A, IEC pub 1000-4-4 level 3
Immunity to Electrical Fast Transients	Impulse Wave	IEC pub 1000-4-5 level 4
Breakdown Voltage	Dielectric Withstand	UL 508, CSA C22.2-14-M1987, IEC pub 1000-4
Immunity to Radiated Fields	RFI	IEC pub 801.3 level 4
Safety	USA	UL 508
Safety	Canada	CSA C22.2-14-M1987
Safety	Europe	VDE/TUV Equivalent to UL508

Table 45 – Agency Compliance

9 System Manager Software Drivers

The ECC will be designed in such a way to be as completely compatible as possible with the existing SMS PowerLogic Network Server MMSTCP and ModbusTCP Drivers. This will allow the integration of the new communications device without major impact on SMS.

REDACTED

10 ECC Operating System

The operating system used in the ECC design will be the pSOSystem operating system from Integrated Solutions, Inc. (ISI). pSOSystem has been established as a leading real-time operating system (RTOS) for embedded applications. pSOSystem is a modular, high-performance, real-time operating system, designed specifically for embedded microprocessors. A few of the main components that will be utilized during development will be the pSOSystem pSOS+ Real-time Multi-tasking Kernel, pNA+ TCP/IP Protocol Stack, pREPC+ ANSI C Standard Library, pROBE+, and pMONT+. All the firmware work will be done in ISI's pRISM+ development environment. For detailed information on these pSOSystem and pRISM+ components, refer to the ISI pSOSystem and pRISM+ documentation.

11 ECC Register List

The ECC will have nearly all configuration data stored in the CM it is inserted into. The required storage size will need to be a minimum of 300 words (registers). The following is the "quick list" of the registers.

Register	HiByte	LoByte
500	IP Address 1st Octet (0 – 255)	IP Address 2nd Octet (0 – 255)
501	IP Address 3rd Octet (0 – 255)	IP Address 4th Octet (0 – 255)
502	IP Sub-net Mask 1st Octet (0 – 255)	IP Sub-net Mask 2nd Octet (0 – 255)
503	IP Sub-net Mask 3rd Octet (0 – 255)	IP Sub-net Mask 4th Octet (0 – 255)
504	IP Router Address 1st Octet (0 – 255)	IP Router Address 2nd Octet (0 – 255)
505	IP Router Address 3rd Octet (0 – 255)	IP Router Address 4th Octet (0 – 255)
506	Ethernet physical connection (0 = UTP, 1 = Fiber HD, 2 = Fiber FD)	
507	ModbusTCP Client Timeout in seconds (5 – 60)	MMSTCP Client Timeout in seconds (5 – 60)
508	HTML Access token expiration time in minutes (1 – 255)	
509	RS485 Timeout in seconds (3 – 10)	DPR Timeout in seconds (3 – 10)
510	CM/RS485 Time Synchronization Interval in seconds (0 = disabled, 30 – 65535)	
511	HTML default language type (0 = English, 1 = French, 2 = Spanish)	
512	RS485 Baud Rate (1200, 2400, 4800, 9600, 19200, 38400)	
513	RS485 Parity (0 = none, 2 = even)	RS485 Mode (1 = 2-wire Smart, 2 = 4-wire Smart)
514	Admin password ASCII character 1	Admin password ASCII character 2
515	Admin password ASCII character 3	Admin password ASCII character 4
516	Admin password ASCII character 5	Admin password ASCII character 6
517	Admin password ASCII character 7	Admin password ASCII character 8
518	Pass1 password ASCII character 1	Pass1 password ASCII character 2
519	Pass1 password ASCII character 3	Pass1 password ASCII character 4
520	Pass1 password ASCII character 5	Pass1 password ASCII character 6
521	Pass1 password ASCII character 7	Pass1 password ASCII character 8
522	Pass1 Password HTML Page Access Bitmap (Most Significant Word)	

REDACTED

523	Pass1 Password HTML Page Access Bitmap (2nd Most Significant Word)	
524	Pass1 Password HTML Page Access Bitmap (2nd Least Significant Word)	
525	Pass1 Password HTML Page Access Bitmap (Least Significant Word)	
526	Pass2 password ASCII character 1	Pass2 password ASCII character 2
527	Pass2 password ASCII character 3	Pass2 password ASCII character 4
528	Pass2 password ASCII character 5	Pass2 password ASCII character 6
529	Pass2 password ASCII character 7	Pass2 password ASCII character 8
530	Pass2 Password HTML Page Access Bitmap (Most Significant Word)	
531	Pass2 Password HTML Page Access Bitmap (2nd Most Significant Word)	
532	Pass2 Password HTML Page Access Bitmap (2nd Least Significant Word)	
533	Pass2 Password HTML Page Access Bitmap (Least Significant Word)	
534	Pass3 password ASCII character 1	Pass3 password ASCII character 2
535	Pass3 password ASCII character 3	Pass3 password ASCII character 4
536	Pass3 password ASCII character 5	Pass3 password ASCII character 6
537	Pass3 password ASCII character 7	Pass3 password ASCII character 8
538	Pass3 Password HTML Page Access Bitmap (Most Significant Word)	
539	Pass3 Password HTML Page Access Bitmap (2nd Most Significant Word)	
540	Pass3 Password HTML Page Access Bitmap (2nd Least Significant Word)	
541	Pass3 Password HTML Page Access Bitmap (Least Significant Word)	
542	RS485 Device Definitions - Protocol (3 = PowerLogic, 8 = Modbus)	RS485 Device Definitions - Address (0 – 254)
543 – 604	62 more registers for up to 62 more Device definitions	
605	Number of viewable defined devices, includes CM that the ECC is attached to (1 – 62)	
606	Web Page Passwords Disabled (Most Significant Word)	
607	Web Page Passwords Disabled (Least Significant Word)	
608	MAC Address Byte 1	MAC Address Byte 2
609	MAC Address Byte 3	MAC Address Byte 4
610	MAC Address Byte 5	MAC Address Byte 6
611	Serial Number Byte 1 (32 bit number)	Serial Number Byte 2
612	Serial Number Byte 3	Serial Number Byte 4
613	Model Number (16 bit number)	
614	Hardware Revision Number Character 1	Hardware Revision Number Character 2
615	Hardware Revision Number Character 3	Hardware Revision Number Character 4
616	Date of Manufacture Month Byte	Date of Manufacture Day Byte
617	Date of Manufacture Year Byte, since 1900	Date of Manufacture Hour Byte
618	Date of Manufacture Minutes Byte	Date of Manufacture Seconds Byte
619 – 638	Scratch Pad/Process Area	
639	Firmware Version Number (2000 = 2.000)	
640	Refresh Rate for Static Tables in seconds (1 – 300)	
641	Force Manufacturing Parameters Flag (AAAA means the ECC excepts the manufacturing parameters from the register list)	
642 – 991	Reserved	
992-998	CM4 Ghost of registers 500 - 506	
999	CM4 Ghost of CM4 RS485 port address	

Table 46 – ECC Register List